



João Manuel Espada dos Santos

Licenciado em Engenharia Informática

Social-Media Monitoring for Cold-Start Recommendations

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : João Magalhães, Prof. Auxiliar,
Universidade Nova de Lisboa

Júri:

Presidente: Professor Doutor João Alexandre Carvalho Pinheiro Leite

Arguente: Professor Doutor Pável Pereira Calado

Vogal: Professor Doutor João Miguel da Costa Magalhães



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2014

Social-Media Monitoring for Cold-Start Recommendations

Copyright © João Manuel Espada dos Santos, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Acknowledgements

First and foremost, I would like to thank my thesis advisor, professor João Magalhães, not only for entrusting me with this thesis but also for his ever-present guidance, trust, flexibility and initiative in helping me push my work a little further. Above all else, I especially want to thank him for his encouragement and positive attitude, which certainly helped me keeping motivated until the very end.

Next, I would like to thank my college, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, for taking me as a dear student and providing some of the best personnel and environment I could wish for, making my life as a student as much joyous as it could have been. More specifically, I would like to thank my department, Departamento de Informática, and especially its various professors from which I had the pleasure to learn, for the great and close relationship with their students and availability outside of their working schedule.

Thirdly, I would like to thank my older colleagues, Filipa Peleja and Flávio Martins, for their contribution to my work and availability to heed to my requests as long as it was physically possible for them. I especially want to thank Filipa not only for sharing some stressful moments but also for the brief and long conversations outside of our working scope, which helped me to feel more confident in asking for help whenever I needed.

I would also like to give a special thanks to my thesis colleagues and friends André Grossinho and Gonçalo Tavares, for sharing this academic year with me very closely, both in the most stressful and in the most relaxed moments, for knowing and expressing interest in my work, for helping me brainstorming in various occasions, for spending a lot of time working by my side, for accompanying me in the so-much-needed breaks and mainly for growing with me as individuals. I would also like to thank my other thesis colleagues and friends, especially Catarina Almeida, Diogo Cordeiro, Hélder Gregório, Hélder Marques and João Claro for sharing this year full intensive work and fun moments with me.

Outside of my academic scope, I would also like to thank all my friends for doing what friends do, especially Ângelo Monteiro and Sara Coelho. To Ângelo not only for the familiar moments, which helped me relax and regain my ground whenever I needed,

but also for being such an active person and motivating me to try and get something out of each day. To Sara Coelho not only for her friendship, but also for sharing my thesis ups and downs, despite not sharing any academic background with me and still trying to understand and show interest in my work. I would also like to specifically thank Andy Gonçalves, Bruno Campino, João Oliveira, Mikail Ribeiro, Patrícia Roque and Vanessa Matos for keeping in touch and for sharing some of the most memorable moments of this last year with me.

I would also like to thank to my whole family for being the coolest family around and for not forgetting me, despite not having the chance of seeing me very often. Special thanks to my god-mother Edviges Guerreiro, for believing in me and watching me from the sidelines, to my grandmother Mariana Esperança for always sharing a smile whenever she sees me, to my cousin Roberto Sequeira for contacting me periodically and sharing his everyday experiences with me and to my other cousins Roberto Guerreiro, Tiago Guerreiro, Catarina Lagos and Elisabete Guerreiro for being present on some of the most iconic moments of my academic life.

Last but not least, to my parents and brother for sharing a home and always watching my back. To my brother Bruno Faustino for always showing interest in my well being and for sharing fun and serious moments with me, to my father Marcos Faustino for taking me in and contributing to my growth more than it would be expected of him and finally to my mother Maria Espada Faustino for doing what mothers do and watching my back, being ever-present and letting me follow my own path without a second thought.

To all of you, thank you so much.

Abstract

Generating personalized movie recommendations to users is a problem that most commonly relies on user-movie ratings. These ratings are generally used either to understand the user preferences or to recommend movies that users with similar rating patterns have rated highly. However, movie recommenders are often subject to the Cold-Start problem: new movies have not been rated by anyone, so, they will not be recommended to anyone; likewise, the preferences of new users who have not rated any movie cannot be learned. In parallel, Social-Media platforms, such as Twitter, collect great amounts of user feedback on movies, as these are very popular nowadays. This thesis proposes to explore feedback shared on Twitter to predict the popularity of new movies and show how it can be used to tackle the Cold-Start problem. It also proposes, at a finer grain, to explore the reputation of directors and actors on IMDb to tackle the Cold-Start problem. To assess these aspects, a Reputation-enhanced Recommendation Algorithm is implemented and evaluated on a crawled IMDb dataset with previous user ratings of old movies, together with Twitter data crawled from January 2014 to March 2014, to recommend 60 movies affected by the Cold-Start problem. Twitter revealed to be a strong reputation predictor, and the Reputation-enhanced Recommendation Algorithm improved over several baseline methods. Additionally, the algorithm also proved to be useful when recommending movies in an extreme Cold-Start scenario, where both new movies and users are affected by the Cold-Start problem.

Keywords: Social-Media, Recommender Systems, Media Monitoring, Sentiment Analysis, Crowdsourcing, Cold-start, Movies, Twitter, IMDb.

Resumo

Gerar recomendações personalizadas de filmes a utilizadores é um problema que geralmente se baseia nas classificações dadas pelos utilizadores a filmes. Estas classificações são geralmente usadas ou para o sistema aprender as preferências de um utilizador ou para recomendar filmes que utilizadores com padrões de classificação semelhantes classificaram muito positivamente. No entanto, sistemas de recomendação de filmes sofrem constantemente do problema do Arranque-a-Frio: novos filmes ainda não foram classificados por ninguém, portanto não são recomendados a nenhum utilizador; de forma semelhante, as preferências de novos utilizadores que não classificaram nenhum filme não conseguem ser aprendidas. Em paralelo, plataformas de Mídia-Social, tais como o Twitter, colecionam grandes quantidades de opiniões de utilizadores sobre filmes, devido a estas se terem tornado bastante populares recentemente. Nesta tese propõe-se explorar as opiniões partilhadas no Twitter para prever a popularidade de novos filmes e mostrar como esta pode ser utilizada para colmatar o problema do Arranque-a-Frio. Adicionalmente, também é proposta a exploração da reputação de directores e actores de novos filmes com o mesmo objectivo. Para avaliar estes aspectos, um algoritmo de recomendação baseado em reputação é implementado e avaliado num conjunto de dados extraído do IMDb com classificações reais de utilizadores a filmes, juntamente com dados do Twitter extraídos entre Janeiro e Março de 2014, para recomendar 60 filmes afectados com o problema do Arranque-a-Frio. O Twitter revelou ser uma boa fonte para prever a reputação de novos filmes e o algoritmo desenvolvido apresentou resultados melhorados quando comparado com várias baselines. Adicionalmente, a abordagem também provou ser útil em casos extremos de Arranque-a-Frio, quando tanto novos filmes como novos utilizadores são afectados por este problema.

Palavras-chave: Mídias-Sociais, Sistemas de Recomendação, Monitorização da Mídia, Análise de Sentimento, *Crowdsourcing*, Arranque-a-Frio, Filmes, Twitter, IMDb.

Contents

1	Introduction	1
1.1	Social-Media Context	1
1.2	Cold-Start in Recommender Systems	2
1.3	Objective	3
1.4	Approach	4
1.5	Contributions	5
1.6	Document Organization	5
2	Related Work	7
2.1	Introduction	7
2.2	Media Monitoring	8
2.2.1	Sentiment Analysis	9
2.2.2	Twitter Monitoring	11
2.3	Content-based Recommendation	13
2.4	Collaborative Filtering	14
2.5	Hybrid Recommendation Techniques	17
2.5.1	Content-Collaborative Recommendation	17
2.5.2	Review-based Recommendation	19
2.5.3	Social-based Recommendation	20
2.6	Crowdsourcing for Social-media	22
2.6.1	Crowdsourcing Systems	22
2.6.2	Crowdsourcing for Sentiment Analysis	24
2.7	Summary	25
3	Measuring Reputations and Popularities in Social-Media	27
3.1	Introduction	27
3.2	Learning the Reputation of Entities	28
3.2.1	Building a Domain-Specific Sentiment Lexicon	29
3.2.2	Building a Linked-Entities Sentiment Graph	31

3.2.3	Computing the Reputation of Entities	32
3.3	Twitter Mining and Classification	34
3.4	Crowdsourcing for Social-Media Ground-Truth	36
3.4.1	Worker Interfaces	37
3.4.2	Worker Qualification	38
3.4.3	Tasks Parameters	39
3.4.4	Tasks Execution	41
3.4.5	Results and Discussion	41
3.5	Summary	44
4	Cold-Start Recommendations	45
4.1	Introduction	45
4.2	Building User Profiles	46
4.2.1	Discovering User Preferences	47
4.2.2	Discovering User Profile Variables	48
4.3	Formal Model	48
4.4	Social-Media Trends and Reputations	51
4.4.1	Popularity of New Movies on Twitter	51
4.4.2	Reputation of Directors and Actors on IMDb	51
4.5	Recommendation with Social-Media Signals	52
4.5.1	Modeling User Preferences with the Reputation of Entities	52
4.6	Summary	54
5	Results and Evaluation	55
5.1	Dataset	55
5.2	Methodology	57
5.3	Results and Discussion	61
5.3.1	Twitter for Estimating Movie Popularity	61
5.3.2	Estimation of the θ_d , θ_a and θ_g Parameters	62
5.3.3	Estimation of the α_t Parameter	62
5.3.4	Influence of User Bias	63
5.3.5	Methods Comparison	64
5.3.6	Cases of Extreme User Cold-Start	66
5.4	Summary	67
6	Conclusions	69
6.1	Summary of Contributions	69
6.2	Challenges and Limitations	70
6.3	Future Work	71

List of Figures

1.1	Example of shared tweets regarding the movie <i>The Great Gatsby</i> (2013).	2
1.2	Comparison of tweets and ratings for <i>Godzilla</i> (2014) after premiere.	3
1.3	Overview of implemented recommendation framework.	4
2.1	Sentiment Analysis Granularity.	10
2.2	Labelled LDA topic distribution for Dimensions (from [RDL10]).	12
2.3	TwitterMonitor architecture (from [MK10]).	12
2.4	<i>k</i> -Nearest Neighbours in Collaborative Filtering.	15
2.5	Example of traditional Crowdsourcing task Interface (from [MT12]).	23
3.1	Timeline for the reputation of new movies, directors and actors.	28
3.2	The Rank-LDA graphical model (from [PSM14a]).	30
3.3	The linked-entities sentiment graph structure (from [PSM14a]).	32
3.4	Example of tweets identified for the new movie "Iron Man 3".	35
3.5	Feature extraction process for tweets referring new movies.	36
3.6	Worker interface for the IMDb sentences task.	38
3.7	Worker interface for the tweets task.	38
3.8	Distribution of Crowdsourcing labels for IMDb sentences.	42
3.9	Distribution of Crowdsourcing labels for tweets.	43
3.10	Twitter classification accuracy by agreement.	44
4.1	Overview of Formal Model computation.	49
4.2	Overview of the complete rating prediction model.	53
5.1	Timeline for Twitter extraction process.	56
5.2	Dataset information on user-movie ratings.	57
5.3	Summary of dataset.	58
5.4	The evaluation methodology.	58
5.5	The evaluated methods.	60
5.6	Twitter-based Movie Ratings vs IMDb Movie Ratings.	61

5.7	Estimation of θ_d , θ_a and θ_g as a function of MAE and F-Measure.	62
5.8	Estimation of α_t as a function of MAE and F-Measure.	63
5.9	Distribution of user bias.	63
5.10	The influence of user bias.	64
5.11	Comparative results for different numbers of watched movies.	65
5.12	MAE and F-Measure results for random recommendations.	66
5.13	Comparative results for user cold-start.	67

List of Tables

2.1	Summary of Review-based Recommendation approaches.	21
3.1	Trial tasks parameters and results.	40
3.2	Main tasks parameters.	41
3.3	Reputation analysis accuracy for 8 popular directors and actors (%).	42
5.1	Overall comparative results.	65



Introduction

This first chapter introduces the context and motivation surrounding the development of this dissertation. First, the Social-Media context is detailed and the problem is identified. The objective of the dissertation and the proposed approach are then briefly summarized. By the end of the chapter, the main contributions are specified and the structure of the document is outlined.

1.1 Social-Media Context

The rising availability of the World Wide Web has resulted in an increased flow of information on online forums and services. Social-Media services, in particular, have been target of tremendously increasing popularity, consequently changing the way people interact with each other. While in past eras opinions and experiences were mostly shared by direct contact, nowadays a great part of this information is shared via Social-Media applications, creating large repositories of data related to all imaginable topics. This fact led Social-Media platforms to become a perfect medium for users to obtain insight into various subjects such as restaurants, events, books and movies. Twitter¹, Facebook² and IMDb³ are only some obvious examples of such platforms. Figure 1.1 shows some examples of shared tweets about the movie *The Great Gatsby* (2013).

In parallel, Recommender Systems were introduced with the goal of generating meaningful product recommendations to users. By filtering the overwhelming amount of existent multimedia information, these are powerful tools capable of alleviating the stress of

¹www.twitter.com

²www.facebook.com

³www.imdb.com

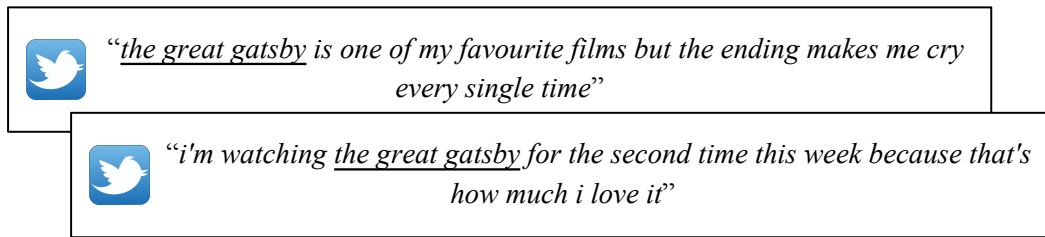


Figure 1.1: Example of shared tweets regarding the movie *The Great Gatsby* (2013).

decision making when it comes to selecting a product, shaping it into a faster and more accurate process. Typical Recommender Systems aim at predicting which products each specific user would like to consume by analysing any useful information it might have on both products and users.

While Social-Media services and Recommender Systems were introduced individually, current state-of-the-art Recommenders rely on Social-Media characteristics to personalize recommendations. IMDb is an example of such Recommenders, where ratings given by users to movies are used to identify users with similar preferences and to recommend movies that similar users to the target user have rated highly.

With Social Networks and Microblogs becoming the trends of this decade, a huge amount of feedback on diverse topics has been collected outside of domain-specific Social-Media platforms. This information, ranging from personal opinions to word-of-mouth suggestions, has the potential to be useful for domain-specific Recommenders. Since the prospective is that Social Networks and Microblogs become increasingly more popular over the next years, exploring this information is certainly a logical and important step on improving domain-specific Recommender Systems.

1.2 Cold-Start in Recommender Systems

Including Social-media feedback on Recommendation Systems is nowadays a standard and most popular Recommenders already include a way for users to discuss and share opinions on items. Two examples are IMDb and Amazon⁴, where users can discuss and opinionate on movies and products, respectively. Collaborative Filtering is the most popular state-of-the-art recommendation algorithm and consists on analysing the historic of consumption of all users in a system to identify similarities in rating behaviours. The identified similarities are then used to predict what to recommend to users who have previously expressed opinions on items, by recommending the products that the most similar users to the target user have rated highly.

While Collaborative Filtering has proven to present excellent results [AT05], relying solely on the feedback given by users propels to the existence of a serious limitation,

⁴www.amazon.com

known as the problem of *Data Sparsity*. This limitation refers to the lack of collected feedback in comparison to the necessary amount to accurately exploit rating behaviours, generally aggravated by the fact that the number of product and items often excels the number of users by a massive margin. Extreme cases of *Data Sparsity* occur for new users and new items: the rating behaviours and preferences of new users who have not rated any product cannot be estimated; likewise, new products that have not been rated by any user cannot be recommended to anyone. These severe cases are the hardest to handle and plague most recommendation approaches, culminating in the most characteristic limitation of Recommender Systems, known as the *Cold-start* problem [MPJ11].

1.3 Objective

Social Networks and Microblogs let users share opinions and experiences of their everyday lives, and the new mobile and wireless technologies let users have access to these at any time and from any place. This ever-present contact with these forms of Social-media leads people into hastily sharing how happy or how disappointed they are with newly obtained products or recently watched movies. Considering how fast Social Networks and Microblogs are flooded with information about new trends, these can actually be a more reliable source of feedback on recently-released products when compared to domain-specific platforms. Figure 1.2 plots a realistic estimation of the accumulated total number of tweets and IMDb ratings shared about the movie *Godzilla (2014)* up to 10 days after its premiere. It can be observed that the shared tweets outnumber the IMDb ratings by a large margin.

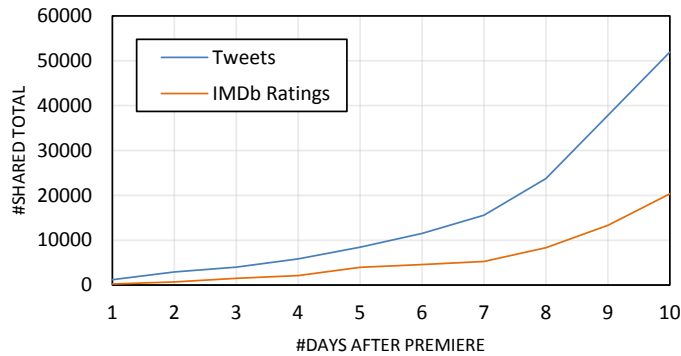


Figure 1.2: Comparison of tweets and ratings for *Godzilla (2014)* after premiere.

This dissertation builds on the idea of exploiting the flux of information on Social-Media platforms to capture feedback on new movies. More specifically, new movies are decomposed into key components, namely the respective directors and actors, whose reputation can be tracked from Social-Media streams. By applying this to a recommendation scenario where new movies lack user-movie ratings and are, therefore, affected by the *cold-start* problem, the objective of this dissertation is defined as:

to develop a recommendation algorithm that explores Social-Media feedback on new movies, directors and actors to recommend movies affected by the Cold-Start problem.

1.4 Approach

New movies that have just been released commonly lack a qualitative measure of their respective quality, as these usually have no user-movie ratings. To tackle this *cold-start* problem, a framework as implemented that mines social-media feedback in order to recommend new movies. The implemented framework can be divided in two main modules: the Social-Media Monitoring module, responsible for mining social-media signals about new movies, and the Cold-Start Recommendation module, responsible for recommending new movies by exploiting the social-media information collected by the other module. Figure 1.3 presents an overview of the described framework.

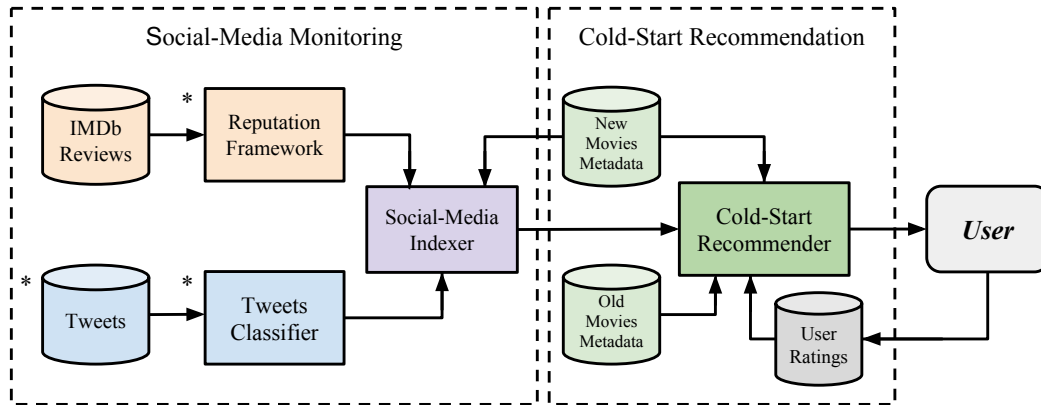


Figure 1.3: Overview of implemented recommendation framework.

The Social-Media Monitoring module comprises two main sets of components: the Twitter Mining components (illustrated in blue) and the IMDb Mining components (illustrated in orange). The Twitter Mining components are responsible for capturing social-media feedback on new movies, affected by the *cold-start* problem: tweets where these movies are identified are stored and labelled according the recognized movie titles. The captured tweets are then classified by a sentiment classifier such that, for each tweet, it is inferred if it is speaking positively or negatively about the identified movies. The IMDb Mining components, in turn, are responsible for computing the reputation of the *cold-start* movies corresponding directors and actors from IMDb user-movie reviews [PSM14a]. Both social-media information (i.e. the tweets about new movies and reputation of the corresponding directors and actors) is indexed together by movie title, in order to allow fast look-ups by the Recommendation module. This module considered the direct contribution of two PhD students: Filipa Peleja [PDM12; PSM14b; PSM14a] contributed with the Reputation Framework [PSM14a] and the Tweets Classifier components; Flávio Martins [MPM12; HMM13] contributed with a dataset of generic tweets, captured in real-time

(refer to Section 5.1 for more information on the dataset).

The Cold-Start Recommendation module comprises the only iteration with a user, i.e. it recommends new movies to users. This module comprises two static databases: one containing the metadata of old, already rated movies, namely their title, directors, actors and genres; other containing the same metadata for the new movies, which are potentially recommended. The metadata of the old movies is used together with the corresponding user-movie ratings to learn user preferences. Oppositely, the metadata of the new movies is used not only for the Social-Media Indexer to related new movies with the corresponding directors and actors, but also for the Recommender to be able to relate user preferences with the new movies. The main component, namely the Cold-Start Recommender, explores the social-media feedback on new movies, their metadata and the discovered user preferences to compute personalized *cold-start* recommendations for the target user.

1.5 Contributions

The goal of this dissertation is to develop, implement and evaluate a recommendation framework that exploits social-media signals to improve recommendation of *cold-start* movie. Furthermore, its main contributions are:

- A Social-Media Monitoring module that captures social-media information on new movies, namely a set of classified tweets referring the movies and the reputation of their respective directors and actors [PSM14a];
- A Cold-Start Recommendation module that recommends *cold-start* new movies to users, by exploring both the users past ratings and the social-media information on the candidate new movies;
- An evaluation of the various components of the implemented modules, performed by leveraging on realistic datasets crawled from IMDb and Twitter;
- Contribution on two scientific papers [PSM14b; PSM14a], published at the 2014 editions of *SIGIR* and *International Conference on Web Intelligence*.

1.6 Document Organization

The rest of the document is organized as follows:

- **Chapter 2:** Related work overview addressing the topics of Media Monitoring, Content-based Recommendation, Collaborative Filtering, Hybrid Recommendation and Crowdsourcing for Social-media.

- **Chapter 3:** Presentation of the media monitoring methods used to capture tweets about new movies and the reputation of its directors and actors. Validation of those methods by using Crowdsourcing practices.
- **Chapter 4:** Presentation of the recommendation algorithm that explores Social-Media feedback to recommend Cold-Start movies.
- **Chapter 5:** Experimental Results and Evaluation of the recommendation algorithm using a crawled dataset with realistic IMDb user-movie ratings and tweets regarding new movies.
- **Chapter 6:** Summary of Final Conclusions, Limitations of the implemented framework and Future Work.



Related Work

This chapter details the most important concepts and background information to the development of this dissertation. First, previous work on social-media monitoring and sentiment analysis is presented. Next, state-of-art recommendation techniques and hybrid recommendation approaches are discussed. By the end of the chapter, crowdsourcing is introduced as a novel process to obtain ground-truth for social-media datasets.

2.1 Introduction

Recommendation Systems were introduced with the goal of generating meaningful item or product recommendations to users. Since different users have different interests, Recommender Systems focus on collecting user information in order to be able to predict what products each specific user would like to consume. Generally, a recommendation technique searches for the n number of products that have the most utility for the target user. Let U be the set of all users, I be the set of all items and $r(u, i)$ be the estimated utility value of item i to the user u . Choosing the item $i' \in I$ that maximizes the user's utility can be formally defined as:

$$i' = \arg \max_{i \in I} r(u, i). \quad (2.1)$$

There are various different approaches to predicting the utility of a not-yet-seen item i to a user u and that's where most recommendation techniques differ from each other. The aim of this dissertation is to implement an hybrid recommendation technique that improves on state-of-the-art approaches when it comes to estimating the utility values of new items, by collecting information shared in Social-media. This chapter presents a

survey of the most relevant concepts and background information to the development of the proposed system.

The most important information in the **Social-media** domain is presented in the following sections:

- **Section 2.2:** Media Monitoring is related to how relevant Social-media information is identified and extracted. This includes an introduction to Sentiment Analysis and a discussion of previous Media Monitoring studies for Twitter.
- **Section 2.6:** Crowdsourcing is related to how algorithms that rely on human-origin data, like Social-media data, are evaluated. This includes an introduction to Crowdsourcing Systems and a discussion of some previous studies using Crowdsourcing for validation of machine learning methods.

Likewise, the most important information on the domain of **Recommender Systems** is presented in the following sections:

- **Section 2.3:** Content-based Recommendation is a state-of-the-art approach to recommendation that relies on the similarity between user and item attributes. This includes a specification of the general algorithm and the discussion of its limitations.
- **Section 2.4:** Collaborative Filtering is a state-of-the-art approach to recommendation that relies on the historic of consumption of users. This includes a specification of the general algorithms and the discussion of its limitations.
- **Section 2.5:** Hybrid Recommendation approaches combine state-of-the-art recommendation techniques with each other or with characteristics of other techniques. This includes a discussion of previously proposed approaches somehow related to the context of the dissertation.

In the end of the chapter is presented a small summary of the most important works in the literature.

2.2 Media Monitoring

The rising availability of the web throughout the world has resulted in an increased flow of information on online forums and services. Media Monitoring is the activity of observing media channels in order to capture this information. When applied to social-media, it can be seen as the process of monitoring *what is being said* on the social web about a determined topic. With the introduction of micro-blogging platforms like Twitter and, to a certain extent, Facebook, users now have a motivation to share any information they feel relevant or interesting with their friends or followers [MCHLM08]. As a result, these platforms emerged as a great source for Media Monitoring studies, since users share all

kinds of information, like real-time news, events or other things of their interest, like movies and books. Twitter, in particular, has been the target of a series of studies and applications with different goals [RDL10; CWNWS12; MK10; LSD12; OSO12].

In the context of this dissertation, Media Monitoring is used to mine and classify movie opinions from Twitter and potentially Facebook, focusing especially on textual micro-posts. Sentiment Analysis techniques are the standard approach to this kind of tasks, since they focus on analysing and classifying the sentiment present on text [PL08]. Therefore, this section focuses mainly on discussing the use of Media Monitoring to mine textual-based sentiment on Twitter.

2.2.1 Sentiment Analysis

Sentiment Analysis refers to the act of computationally identifying and extracting subjective information from textual sources, such as opinions towards a given item [PL08]. Generally speaking, it aims to identify the context or feelings expressed by the writer of a document. Due to the increasing popularity of the Web, the amount of textual information shared online has grown immensely, leading Sentiment Analysis to be a research area of respectable attention in the last decade.

The most basic form of Sentiment Analysis consists in identifying the sentiment polarity of a piece of text: whether the text contains mostly *negative*, *neutral* or *positive* sentiment. An example of this was presented by Pang et al. in [PLV02], where sentiment analysis is used to identify the overall sentiment polarity of movie reviews. The same author later presented a similar study in [PL05], where sentiment polarity is identified and represented in a rating scale from 1 to 3, with 1 representing the *most negative* polarity and 3 representing the *most positive*. In this study, Pang argued that a rating scale is more precise in representing degrees of sentiment versus the simplified representation of *positive* or *negative* since, for a larger scale, it is possible to represent more levels of *positivity* and *negativity*. Sentiment Analysis also considers other tasks, like identifying emotional states. In [SM08], Strapparava et al. presented a method to extract emotions from text, namely *anger*, *disgust*, *fear*, *joy*, *sadness* and *surprise*.

Sentiment Analysis goes beyond identifying sentiment polarity and emotional states. *Subjectivity vs Objectivity* identification is a finer-grain and more complex form of Sentiment Analysis tasks [MBW07]. These tasks consist of distinguishing subjective text from objective text: in other words, distinguishing *opinions* from *facts*. While sentiment polarity is often associated to document-level analysis, *subjective/objective* identification is often associated to sentence-level analysis. Pang et al. argue in [PL04] that removing *objective* sentences from Documents improves results of sentiment polarity analysis. In a finer level of granularity resides *Aspect-Based Sentiment Analysis*, which consists in identifying sentiments expressed towards certain attributes of entities [PL08]: for example, identifying the opinion towards the monitor or the processor of a laptop. In [JO11], Jo et al. presented an example of this level of granularity by using Sentiment Analysis to obtain

features of electronics and restaurants. Figure 2.1 summarizes the main tasks characteristic of each granularity level.

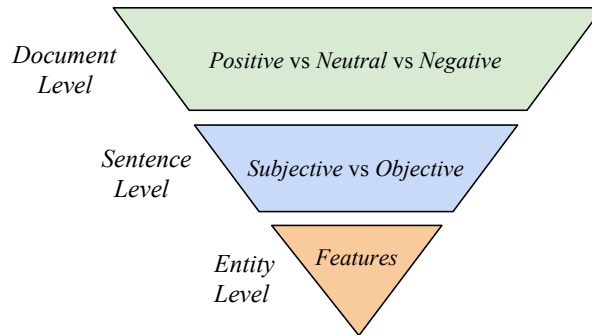


Figure 2.1: Sentiment Analysis Granularity.

A Sentiment Analysis task can be viewed as a two-steps process: first, the text is analysed and the words that have any sentiment value are identified and extracted; then, a classifier is used with the set of extracted sentiment words to calculate the overall polarity of the text. The standard techniques to identify the sentiment words can be split in the following approaches [PL08]:

- **Manual Approach:** Manually identifying the sentiment words. While it is trivial for humans to identify sentiment words, it's also highly time consuming, so this approach is commonly combined with other automated methods [DC01].
- **Corpus-based Approach:** Relies on universal facts about how words are used and linguistic rules to exploit co-occurrence patterns within the corpus to identify the sentiment words [Tur02].
- **Dictionary-based Approach:** A Dictionary containing the sentiment words and respective polarity is used. Words are normalized and extracted if they are found on the dictionary. A popular linguistic dictionary is the *SentiWordNet* [ES06] .

Obtained Sentiment Words can have various representations. The most basic representation is the single word, also known as *unigram*. According to Pang et al. in [PL04], the *unigram* representation presents fairly good results. Sentiment Words can, however, be represented as N words: those representations are called the *N-grams* and are able to capture sequences of words that would otherwise not be captured. After the Sentiment Words are extracted from the text in the desired representation, a classifier is used to obtain the overall polarity. Variants of *Naïve Bayes* and *Support Vector Machines* are some classic classifiers [WM12].

While it might be trivial for humans to understand the sentiments present on text, for artificial intelligence it is still a real challenge, even for the most sophisticated computers [YCK09]. Considering that, there is no reliable unsupervised computational method

to evaluate the accuracy of Sentiment Analysis techniques. Therefore, the accuracy of a method is generally obtained by how well it agrees with human judgement. A usual approach is resorting to Crowdsourcing practices to obtain human-origin labels on analysed documents. Crowdsourcing practices are further discussed in section 2.6.

2.2.2 Twitter Monitoring

As a social networking service on an online social era, Twitter has gained great success in recent years [LSD12] and has been the target of a series of Media Monitoring-based studies and applications, focused on shared textual micro-posts.

Chen et al. studies, in [CWNWS12], the application of sentiment analysis to identify the sentimental polarity and domain corpus of unlabelled tweets. The author discusses that, due to the short length of tweets and the abundance of slang words, traditional sentiment analysis approaches are not the best to accurately pursue the desired task. As an alternative, the author proposes a method that first collects a dictionary of both formal and slang words: formal from *SentiWordNet* and slang from *Urban Dictionary*. After that, the method identifies the corpus's candidate expressions found on tweets and builds two networks to represent dependencies between words. The networks are then used to estimate the polarities of candidate expressions using a probabilistic optimization model. By performing experiments on two domains (*people* and *movies*), the author shows that the approach greatly improves the performance when compared to several baseline methods, both in terms of accuracy and scalability.

In [OSO12], Ozdakis et al. studies the usage of *hashtags* to improve event detection in tweets, when compared to standard word-based vector generation methods [GM05]. In his study, Ozdakis uses a clustering algorithm to agglomerate tweets according to their similarity in vector space model, applied to four different generation methods in order to compare the results: using words in tweets without semantic expansion; using words with semantic expansion; using *hashtags* without semantic expansion; using *hashtags* with semantic expansion. The obtained results show that using *hashtags* improves accuracy of event detection for Twitter, and the application of semantic expansion further improves the method. Additionally, experiments including *hashtags* detected event fast, suggesting improved temporal performance.

In [RDL10], Ramage et al. presents a partially supervised model (a labelled alternative of LDA [BNJ03]) to map the content of tweets into dimensions, corresponding roughly to substance, style, status and social characteristics of posts. The proposed model, Labelled LDA, differs from the standard model in the sense that labels can be introduced to a subset of posts so the model can learn sets of words that go with particular labels. Considering the Twitter structure, one example is using *hashtags* as labels. Experiments were performed with the help of manual labelling to identify the dimensions of the different obtained topics, evidencing that the method can support rich analysis of Twitter content at a large scale. The author discusses that successfully mapping topics into dimensions

can be useful to personalize feeds as, if implemented, permits users to filter dimensions in which they have no interest. Figure 2.2 presents some example topics identified for each dimension.

Category	Fleiss' κ	Example topic
Substance 54/200	.754	obama president american america says country russia pope island failed honduras talks national george us usa
Status 30/200	.599	am still doing sleep so going tired bed awake supposed hell asleep early sleeping sleepy wondering ugh
Style 69/200	.570	haha lol :) funny :p omg hahaha yeah too yes thats ha wow cool lmao though kinda hilarious totally
Social 21/200	.370	can make help if someone tell_me them anyone use makes any sense trying explain without smile laugh
Other 47/200	.833	la el en y del los con las se por para un al es una su mais este nuevo hoy

Figure 2.2: Labelled LDA topic distribution for Dimensions (from [RDL10]).

On the applicational realm, Mathioudakis et al. proposed a framework in [MK10] to detect real-time trends by monitoring Twitter streams. The framework, called *TwitterMonitor*, exploits co-occurrences of *Bursty Keywords* (keywords that are encountered at an unusually high rate at a certain time frame) to identify trends as groups of keywords. Then, by employing context and entity extraction algorithms to the tweets of the detected trends, the framework composes textual descriptions as identifiers for the detected groups of keywords. As an hypothetical example, a trend composed by the keywords "Depp" (from the actor Johnny Depp) and "Nolan" (from the director Christopher Nolan) could be related to (and identified as) the "Movie Oscars". Figure 2.3 presents the architecture of *TwitterMonitor*, where tweets are mined with the help of a *Stream Listener* specific to the *Twitter API*.

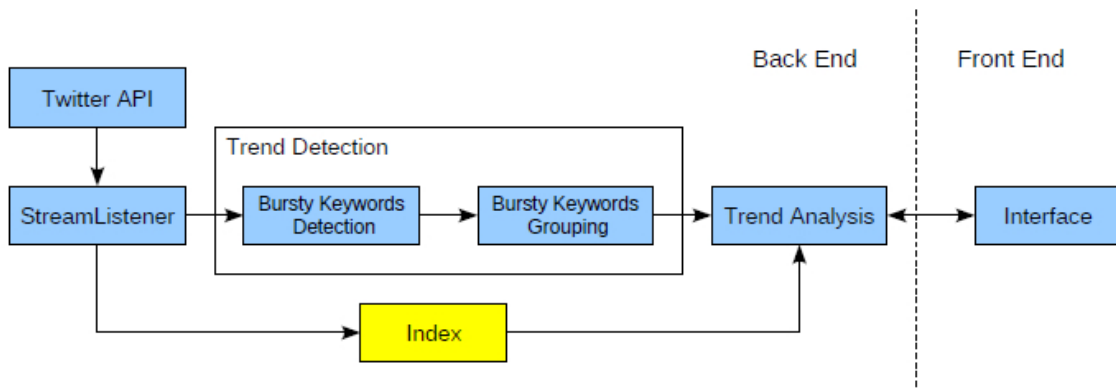


Figure 2.3: TwitterMonitor architecture (from [MK10]).

More recently, Li et al. proposed a similar framework in [LSD12] called *Tweevent*, with

the goal of detecting events from *bursty* tweet segments. In *Twevent*, identified *bursty* segments of tweets are described by the tweets containing that segment in a determined time frame. The identified segments are then clustered according to the similarity on their respective tweets, resulting in candidate events. In the end, Wikipedia entities are used to filter candidate events, resulting in a set containing the realistic and most newsworthy events. In experiments performed by the author, *Twevent* demonstrated great results, especially in distinguishing relevant events from the noisy ones.

2.3 Content-based Recommendation

Pure content-based recommendation consists in recommending items to users based on their similarity, using content-based filtering techniques to qualify the harmony of user-item relationships. As the name suggests, content-based filtering techniques rely on analysing the content data of both items and users to infer their similarity. Since these techniques have its roots in information retrieval [BC92] and text processing [Sal89], the content data is usually textual.

The most common approach assumes that users and items are represented in a non-atomic manner, where each is but a collection of attributes, also known as *keywords* [AT05]. These can have different meanings depending on the purpose and context of the system. In a movie recommender like IMDb, *keywords* can include genres and names of actors and directors: while these represent characteristics for items (movies), they represent preferences for users. The set of attributes that represent an entity is called that entity's *profile*. Let $K = \{k_1, k_2, \dots, k_n\}$ be the dictionary of existing *keywords*. In a general context, the *profile* of user u_i can be formally represented as the vector

$$u_i = (w_{i1}, w_{i2}, \dots, w_{in}), \quad (2.2)$$

where w_{in} represents the weight, or importance, of the *keyword* k_n to the user u_i . The representation of an item's *profile* follows the same definition.

Characterizing a user or an item as a set of *keywords* can be done implicitly or explicitly: in any case, this process is called *profiling*. For items, this process is often performed implicitly by analysing item data directly, like its description and content. For users, implicit *profiling* generally consists in analysing the properties of previously consumed items, while explicit characterization is often achieved by asking the user to describe its preferences manually.

Considering how entities are usually represented in these systems, inferring their similarity means calculating the proximity between their profile vectors. Many similarity measures to describe the proximity of two vectors exist, but the cosine similarity is the most widely used [LGS11]. Therefore, the estimated utility value $r(u_i, i_j)$ of the item i_j to

the user u_i is usually obtained as:

$$r(u_i, i_j) = \frac{\sum_k w_{ik} w_{jk}}{\sqrt{\sum_k w_{ik}^2} \sqrt{\sum_k w_{jk}^2}}. \quad (2.3)$$

While content-based recommenders present great results overall, they also present several limitations. Since these techniques rely solely on analysing the metadata associated to entities to make recommendations, there is the problem of limited content analysis: not only it is necessary for entities to have enough metadata to create a reliable *profile*, but the content also needs to be in a format that is automatically parsable by a computer. While information retrieval techniques work well for textual data, the same cannot be said about other types of data like multimedia, making it hard to extract characteristics from content like images, audio streams and video streams. Manually assigning *keywords* is also not a possibility to describe items due to the limitations of resources. This problem is also associated to the fact that *keywords* only describe attributes and not the quality of the item, resulting on the algorithm recommending products with characteristics the user likes but not necessarily high quality products. Another issue that is implicitly related to limited content analysis is the *new user problem*: while the system doesn't have enough data to make a reliable user *profile*, it is impossible to guarantee the quality of recommendations for that user.

The most characteristic limitation of content-based filtering systems is a problem known as *overspecialization*. This problem refers to the tendency of content-based recommenders to exclusively recommend items with similar attributes to the user, leading to recommendations that are neither varied nor unexpected. Per example, an user that never watched a romance movie will never receive a recommendation for even the greatest romance movie ever released. The use of genetic algorithms to introduce some randomness has been proposed as one of the approaches to address this issue [SM93]. In some cases, it is also not desirable that similar items are recommended, such as various different articles regarding the same event. To avoid this, some content-based recommenders, such as Daily-Learner [BP00], filter items that are too similar to previously consumed by users, guaranteeing recommendation diversity and reducing redundancy.

2.4 Collaborative Filtering

Collaborative Filtering consists in analysing the consumption history of all users in a system to identify patterns in rating behaviours, exploiting those patterns to predict what to recommend to a user who has previously rated items. These techniques have its roots in 1992, when Goldberg et al. proposed an e-mail filtering system where users could contribute with feedback about the content of e-mails, resulting in a collaborative effort to sort e-mails in terms of relevance [GNOT92]. According to Adomavicius et al. in [AT05], Ringo [SM95], Video Recommender [HSRF95] and GroupLens [KMMHGR97] were the first systems to use collaborative filtering to automate prediction.

The most common approach to collaborative filtering in recommenders rely on explicit feedback, most commonly numeric ratings. An example of this kind of feedback is implemented by IMDb, where users can rate movies on a numeric scale ranging from 1 to 10, where 1 represents the worst opinion on quality and 10 the best. Other forms of explicit feedback can also be used, like written reviews, but numeric ratings are the most popular format since they are easier to obtain and compute. Oard et al. discussed in [OK+98] the use of implicit feedback obtained by observing user behaviours, like considering the purchase of an item as positive feedback. Although possibly useful, explicit evaluation is still more reliable since it reflects a preference intentionally provided by the user. In any case, feedback is usually represented as an User-Item Matrix.

Collaborative Filtering algorithms can be grouped in two classes: *Neighbourhood-based* (also referred to as *Memory-based*) and *Model-based*. In *Neighbourhood-based* algorithms, recommendation predictions for a user u_a are based on the feedback given by the k users identified as the most similar to the user u_a . The k most similar users are generally obtained by applying the *k-Nearest Neighbours Algorithm* [ESK03] to the rating matrix: this algorithm computes a matrix of similarities between all users and identifies the k most similar to the target user u_a (Figure 2.4).

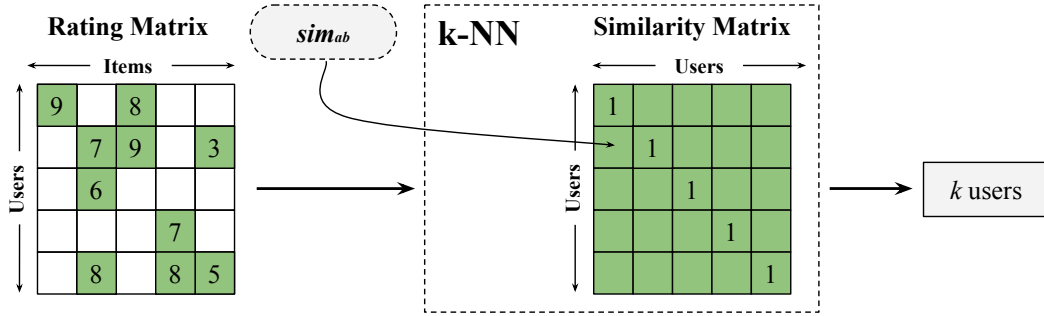


Figure 2.4: *k-Nearest Neighbours* in Collaborative Filtering.

Many similarity measures to describe the similarity between two users exist, but the Pearson Correlation Coefficient [BCHC09] is the most widely used. Therefore, the similarity value sim_{ab} between target user u_a and a user u_b is usually obtained by:

$$sim_{ab} = \frac{\sum_{i \in I} (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b)}{\sqrt{\sum_{i \in I} (r_{ai} - \bar{r}_a)^2 \sum_{i \in I} (r_{bi} - \bar{r}_b)^2}}, \quad (2.4)$$

where I is the set of items rated by both u_a and u_b , r_{ai} is the rating given by u_a to item i , r_{bi} is the rating given by u_b to item i , \bar{r}_a is the average rating given by u_a and \bar{r}_b is the average rating given by u_b . After the k users are selected, the estimated utility value $r(u_a, i_i)$ of an item i_i to the user u_a can be obtained by the weighted average of their feedback:

$$r(u_a, i_i) = \bar{r}_a + \frac{\sum_{b \in K} (r_{bi} - \bar{r}_b) \cdot sim_{ab}}{\sum_{b \in K} sim_{ab}}, \quad (2.5)$$

where K is the set of the k most similar users. When applied to millions of users, the

described approach doesn't scale well due to the complexity of the search for the similar users. In [LSY03], Linden et al. proposed an analogous alternative that matches u_a 's rated items to similar items, leading to faster and even improved recommendations. Similarly to the standard approach, the item-centric alternative starts by calculating item similarity with the Pearson Correlation Coefficient. After the k similar items are selected and stored in K , the estimated utility value $r(u_a, i_i)$ of an item i_i to the user u_a can be obtained by the simple weighted average,

$$r(u_a, i_i) = \frac{\sum_{j \in K} r_{aj} \text{sim}_{ij}}{\sum_{j \in K} |\text{sim}_{ij}|}. \quad (2.6)$$

In contrast to *Neighbourhood-based* approaches where the rating matrix is used directly to compute rating predictions, *Model-based* algorithms use the rating matrix as training data to learn a model, which is then used to make the predictions. The underlying idea of *Model-based* approaches is that some unspecified factors that should be considered are not captured in the rating matrix, like user bias and temporal fluctuations. Considering that, the idea is to *get to know the user* in order to predict how he would rate an item, instead of recommending what similar users liked. These approaches are heavily influenced by machine learning techniques and exist in great variety, including Bayesian Network Models [Jen96], Clustering Models [UF98] and Latent Semantic Models [Hof04]. While these surpass *Neighbourhood-model* algorithms in terms of accuracy, the computational complexity of building the model makes them natively slower.

An example of a *Model-based* algorithm is presented by Breese et al. in [BHK98], where the utility value of an item i_i to a user u_a is given by:

$$r(u_a, i_i) = \sum_{r \in R} r \cdot Pr(r_{ai} = r | r_{ui'}, i' \in I_u), \quad (2.7)$$

where R is a set of all possible ratings, and $Pr(\dots)$ is the probability of user u_a giving a rating of r to item i_i , considering previously rated items. The probability $Pr(\dots)$ is obtained from a previously constructed model using either Bayesian Networks or Clustering Algorithms.

While resorting only to the rating structure makes pure Collaborative Filtering systems simple and accurate, that property also highlights some limitations. Like Content-based systems, Collaborative Recommenders also suffer from the *new user problem*: since recommendations are made based on what a user has rated, if a user haven't rated anything it's impossible to recommend any item accurately. Analogously, items that haven't been rated by any user will never be recommended: this problem is known as the *new item problem*. Both these problems contribute to the most characteristic limitation of Collaborative Filtering systems, known as the problem of *Data Sparsity*. This problem refers to the lack of collected ratings compared to the number of necessary ratings to obtain the best accuracy possible: there's usually a greater number of unrated items than of rated ones. For example, in a movie recommender, there may be movies with a small number

of ratings that would never be recommended, even if their rating average is very high. Some approaches to reduce the impact of this limitation have been previously suggested, like the application of *Demographic Filtering* [Paz99] and the use of the *Singular Value Decomposition* technique to reduce the dimensionality of sparse rating matrices [SKKR00].

2.5 Hybrid Recommendation Techniques

Hybrid Recommendation emerged with the goal of improving state-of-the-art recommendation techniques, namely pure Content-based Recommendation (Section 2.3) and pure Collaborative Filtering (Section 2.4). While each of the referred family of techniques presents great results on its own, each also presents several limitations, leading research on recommendation systems to focus on hybrid methods to tackle those issues [Bur02]. Several variations of hybrid recommendation have previously been proposed, ranging from exclusive combinations of Content-based Recommendation with Collaborative Filtering to the incorporation of external techniques like Sentiment Analysis and Media Monitoring. Some of the proposed approaches are discussed in this section, presented under the following categories:

- **Content-Collaborative Recommendation:** Combines characteristics of both Content-based and Collaborative approaches. Discussed in 2.5.1.
- **Review-based Recommendation:** Combines Sentiment Analysis techniques with pure Recommendation approaches. Discussed in 2.5.2.
- **Social-based Recommendation:** Applies Social knowledge, theories or context to pure Recommendation approaches. Discussed in 2.5.3.

2.5.1 Content-Collaborative Recommendation

Content-Collaborative Recommendation methods combine characteristics of both Content-based Recommendation and Collaborative Filtering in order to leverage the strengths of each technique. According to Adomavicius et al. in [AT05], the different ways in which the techniques can be combined are:

- By implementing both approaches separately and combining their respective predictions in different ways.
- By incorporating Content-based Recommendation characteristics into Collaborative Filtering, in order to attenuate Collaborative Filtering limitations.
- By incorporating Collaborative Filtering characteristics into Content-based Recommendation, in order to attenuate Content-based limitations.
- By implementing a general unifying model that incorporates characteristics of both techniques.

The first and most straight-forward approach to Content-Collaborative Recommendation enunciated consists on implementing both approaches separately and combining their respective predictions. Cotter et al. presented, in [CS00], a personalized television guide that recommends items by using Content-based and Collaborative Filtering methods independently and merging the resulting lists into a final list. The author demonstrates that the method ensures recommendation diversity, by the use of collaborative filtering, and solves the latency problem, by using the Content-based counterpart. Claypool et al. presents a similar method in [CGMMNS99], applied to an online newspaper, where the two prediction lists are combined by an weighted average of the obtained utility values for each item. A different alternative is applying both Content-based and Collaborative Filtering but only selecting the list with best results, following a specified criterion. DailyLearner [BP00] is an example of this alternative, where news are selected by the recommendation technique that produces recommendations with the higher level of confidence. Another example is presented in [TC00], where commercial items are recommended based on the technique that produces results that are more consistent with the past ratings of the user.

The second Content-Collaborative approach enunciated consists in attributing pure Content-based characteristics to Collaborative Filtering. Pazzani et al. describes in [Paz99] an approach to Collaborative Filtering where the most similar users are obtained by calculating the similarity between content-based *user profiles*, instead of the rating matrix typically used in collaborative systems. In this approach, the *profiling* process for a user is done by applying the Winnow algorithm [Lit88] to a set of items positively rated by the user, used as training data. A similar method is used by Fab [BS97], where web pages are recommended not only when they rank well against a *user profile*, but also when they rank well against the most similar users. According to Pazzini in [Paz99], mixing a content-based *user profile* into a collaborative effort allows the systems to overcome some sparsity-related issues, especially in cases where users have similar tastes but didn't happen to rate the same specific items. Additionally, items are not only recommended when they have been positively rated by similar users, but also when they score well against the user's preferences. An example of this Content-Collaborative approach in the movies domain is presented by Melville et al. in [MMN02], where a framework is proposed to collaboratively recommend movies after enhancing the rating matrix with a Content-based predictors. This content-based predictor is an extended Bayesian text classifier that predicts the rating by calculating the similarity harmony of user-item relationships from their respective *profiles*. Melville et al. also demonstrates that the proposed framework preforms better than pure Collaborative Filtering, pure Content-based Recommendation and a linear combination of both.

Due to Collaborative Filtering being generally preferred and overall presenting better results than Content-based Recommendation, not many varieties of Hybrid Content-based techniques are popular [AT05]. According to Adomavicius et al. [AT05], the most popular approach to these resides in using dimensionality reduction techniques on a

group of content-based *profiles* [AT05]. Soboroff et al. presents an example in [SN99], where *latent semantic indexing* is used to create a collaborative view of a set of *user profiles*, resulting in a merged profile representing the whole group. The author demonstrates that the method presents improved results when compared to pure content-based recommendation.

Finally, some Content-Collaborative approaches divert greatly from the pure methods, while still incorporating aspects of both. Basu et al. presents, in [BHC+98], a method based on *rule induction* that uses rules (as in logic) to predict if a user likes or dislikes a movie. This method uses content-based characteristics since rules are related to item and user features (like the genre). The method incorporates collaborative characteristics by including rules that can relate different users, as exemplified by the rule *users who liked movies of genre X*, being *X* a valid genre. In [PPL01], Popescul et al. presents another unique Content-Collaborative approach, where the Probabilistic Latent Semantic Analysis technique presented in [Hof99] is extended to incorporate three dimensions of data, nominatively *users*, *items* and *item content*. This approach is also presented to tackle *data sparsity* quite effectively.

Overall, all presented Content-Collaborative methods address the issues of *data sparsity* and *overspecialization*: by incorporating Content-based characteristics the methods tackle issues related to the sparsity of the rating matrix, while incorporating Collaborative characteristics contributes to recommendation diversity. Additionally, the content-based characteristics guarantee a certain immunity to the *new item problem*, since items always have features and don't depend solely on acquired ratings. The main weakness of these hybrid methods fall on the *new user problem*, since neither pure Content-based nor pure Collaborative characteristics tackle this problem effectively.

2.5.2 Review-based Recommendation

One recently discussed issue in state-of-the-art recommendation algorithms is the focus on the specific metrics (rating matrix and user profiles) to make recommendations, disregarding the valuable information expressed on free-text reviews [JWMG09]. This form of feedback is rich in information, since users nowadays comment on mostly everything and several web applications, like online forums, only support textual feedback. Review-based Recommendation methods combine Sentiment Analysis techniques with pure recommendation to capture this information, in order to expand the array of considered metrics when recommending items.

Moshfeghi et al. proposes, in [MPJ11], an approach to recommending movies that considers not only the rating matrix, but also emotions and semantic spaces to better describe items and users, in order to tackle *data sparsity* and the Cold-start problem. In this approach, Sentiment Analysis is used to obtain emotion spaces from user reviews and plot summaries. When processing recommendations, the Latent Dirichlet Allocation [BNJ03] is used to compute a set of latent groups of users based on the identified spaces.

The author demonstrates that the proposed method yields great results and improves upon the proposed limitations.

Still in the movies domain, Jakob et al. explores, in [JWMG09], the advantages of improving Collaborative Filtering by also considering the sentiment extracted from user textual reviews. In his approach, opinion mining techniques are first used to identify movie aspects in user reviews, which are clustered by different topics. By observing the overall rating associated to a certain clustered topic given by a certain user, the recommender can learn what are the most important topics for a certain user and recommend movies accordingly. The method improves upon the pure Collaborative Filtering techniques by reducing the number of false positives in recommendations.

In [LCC06], Leung et al. suggests enhancing the rating matrix in Collaborative Filtering by inferring numeric ratings from textual user reviews. In his study, the author proposes a new method to identify sentiment words, semantic orientation and corresponding sentimental strength. The method is sensible to various domains and allows words to have different sentimental orientations: for example, the word *frightening* has a negative orientation in a general context, but might have a positive orientation when used in movie reviews. While, theoretically, enhancing the rating matrix with ratings inferred from textual user reviews tackles *data sparsity*, the author didn't perform any evaluation on the recommendation part.

More recently, Zhang et al. proposed, in [ZDCL10], an approach to a sentiment-based recommender on an online video service that extracts a *like / don't like* rating from reviews and comments to the video description through a *keyword* vector space model. The presented framework uses a Content-Collaborative recommendation approach where the extracted ratings are used to associate *keywords* to an user matrix and an item matrix, later used in combination with the rating matrix to generate recommendations.

Lastly, in [AZSD07] Aciar et al. proposes recommendation from textual reviews by using an ontology to translate the reviews text. The author's ontology relies not only on the reviews positive and negative orientation, but also on the user's skill level. Interestingly, the proposed ontology is able to co-relate product characteristics: for example, on the photographic cameras domain the concept *carry* is related to the feature *size* [AZSD07]. The utility value of an item to an user is calculated by measuring the quality of the various item features.

Overall, the studied Review-based Recommendation techniques vary greatly in properties, even if all tackle *data sparsity* in some particular form. Table 2.1 summarizes the presented methods in relation to Collaborative Filtering properties and specifies the methods that explicitly tackle *data sparsity*, therefore attenuating the Cold-start problem.

2.5.3 Social-based Recommendation

The emerging popularity of Social Web has raised new application areas for recommender systems [GGMS13]. More specifically, Social Networks like Facebook and Twitter have

Method	Requires Rating	Collaborative F.	Handles Sparsity
Moshfeghi [MPJ11]	yes	no	yes
Jakob [JWMG09]	yes	yes	no
Leung [LCC06]	-	yes	-
Zhang [ZDCL10]	yes	yes	yes
Acıar [AZSD07]	no	no	no

Table 2.1: Summary of Review-based Recommendation approaches.

brought the idea of online user relationships to the foreground, demanding research on recommender systems to consider the potential use of these relationships to improve recommendation. Not only that, but even online discussion forums have been more active than ever in recent years, since Social Web applications motivate users to share opinions and discuss points of view. In the scope of this document, Social-based Recommendation refers specifically to hybrid recommendation approaches that are aware of user relationships or social data external to the system, obtained from Social Forums.

In [ALPKO09], Amatriain et al. proposes the usage of "expert" opinions, extracted from an Online Forum accepted as a trust-worthy source, as the basis for recommending movies using an alternative Collaborative Filtering approach. The author proposes that the *k-nearest neighbours* algorithm is slightly modified so it generates a matrix containing the similarity of all users with the experts, instead of the similarity between all users. Consequently, the method predicts recommendations by comparing the target user rating patterns only with experts rating patterns. Experiments on this approach demonstrate that the algorithm attenuates *data sparsity* and the Cold-start problem. The reason for this result is that globally-accepted "experts" online forums are more likely to have rated great numbers of movies.

Some recently proposed hybrid approaches that consider user relationships have also been briefly studied. In [ASS13], Alexandridis et al. proposes to improve recommendation diversity by not only considering similar users to the target user, but also explicitly related users on the target user's social network, referred to as *friends*. The main idea is that there is a certain implicit similarity factor between related users and, by exploring the past consumptions of *friends*, it is possible to find some user interests that are not explicitly obtained by analysing his consumption history. This approach also includes random walks and recommends items by clustering the consumption of items. The author considers the results of the method to be satisfactory as a starting point for further improvement. Another method is proposed by Burke et al. in [BV13], where a Collaborative tagging approach is used as the basis of music recommendation. In this approach, the tags attributed by users to songs are used to create a network of transitive similarities, expanding recommendation diversity. Additionally, the hybrid technique also considers overall social popularity of songs, that potentially tackles the *new user problem*. Primary

results demonstrate that the method presents great results, when compared to other state-of-the-art tagging-based recommenders.

2.6 Crowdsourcing for Social-media

Social-media contexts allow users to share personal information like opinions, statuses and ideas. Although virtual, these contexts are still social and, as such, usually deal with emotions and common sense. Content of this nature might be trivial for the human mind to process and understand, but for artificial intelligence it is still a real challenge, even for the most sophisticated computers [YCK09].

Crowdsourcing is defined as the process of remotely obtaining needed services and contributions from large groups of people over the Web, like online communities. More formally, it can be considered a distributed problem-solving and scalable production model, as it is capable of dealing with large bulks of tasks due to the huge amount of people accessing the Web everyday [YKL11a]. Considering this, Crowdsourcing emerges as the perfect model to solve tasks related to Social-media data. An example of informal Crowdsourcing practices is *Yahoo! Answers*: a general question-answering forum where users request answers or opinions for a given question, and anyone can contribute.

With the increasing popularity of Social Networks, these have become a repository of human-origin data that can be useful for a variety of commercial, academic and social studies. These kind of studies require more formal and supervised hosts for publicising tasks, when compared to examples like *Yahoo! Answers*, in order to guarantee a certain degree of quality for the obtained results. In the context of this document, these hosts are classified as Crowdsourcing Systems.

2.6.1 Crowdsourcing Systems

Crowdsourcing Systems are usually presented as websites designed specifically for Crowdsourcing labour. Some popular examples include *Amazon Mechanical Turk*, *CrowdFlower*, *Taskcn* and *TopCoder* [YKL11b].

Users of these websites can either be requesters or workers: requesters submit jobs and workers perform submitted jobs. Most Crowdsourcing websites contain a framework to help requesters create simple interfaces for their jobs. A traditional interface for a Crowdsourcing job usually considers three components:

- **Data:** Presents the data that the worker is supposed to "evaluate".
- **Question:** Presents a question about the data to evaluate. In other words, how the requester wants the worker to evaluate the presented Data.
- **Answer Field:** Presents a field for workers to submit their answer to the question.

A task generally contains various units of data. For each unit of data, a task can include various questions and, consequently, various answering fields. Generally speaking, a question is a way to ask workers to perform judgement on data. Depending on the objective of the study, the following types of judgement can be requested [TMM13]:

- **Binary Relevance:** When a unit is either relevant or not. *True vs False* questions are an example of this type of judgement.
- **Multi-level Relevance:** When a unit has various levels of relevance. For example, rating a movie on a rating scale of 1 to 10.
- **Ranked Relevance:** When units are compared to others. For example, ordering various movies in terms of quality.

Figure 2.5 presents an example of a traditional interface for a Crowdsourcing task. The figure illustrates one unit of data (a prompt word) and two questions (what is the word with closest meaning; how positive is the prompt word). The requested judgements are both examples of *Multi-level Relevance*.

Prompt word: *startle*

Q1. Which word is closest in meaning (most related) to *startle*?

- *automobile*
- *shake*
- *honesty*
- *entertain*

Q2. How positive (good, praising) is the word *startle*?

- *startle* is not positive
- *startle* is weakly positive
- *startle* is moderately positive
- *startle* is strongly positive

Figure 2.5: Example of traditional Crowdsourcing task Interface (from [MT12]).

Crowdsourcing Systems are appropriate and reliable for formal studies, as they usually contain various mechanics to filter untrustworthy workers. Not only that, these also generally require requesters to reward workers monetarily for their work, motivating workers to perform various jobs wholeheartedly. While workers are not guaranteed to have any expertise on a task's research field, studies by Snow et al. [SOJN08] and Nowak et al. [NR10] showed that an average of 4 non-expert judgements per unit emulate expert-level judgement quality. Not only that, but various techniques to improve the quality of obtained results have already been studied: for example, Sheng et al. demonstrated that repeated labelling can improve the quality of results at a low cost, especially with noisy labels [SPI08].

2.6.2 Crowdsourcing for Sentiment Analysis

On the academic context of Computer Science, formal Crowdsourcing is usually used to obtain the Ground-truth for datasets of human-origin data. Ground-truth refers to the *true value* of a piece of data, and is usually used as benchmark to validate the accuracy of machine learning methods, such as Sentiment Analysis techniques.

In [MT12], Mohammad et al. used Crowdsourcing to generate a sentiment lexicon for sentiment analysis tasks via *Amazon Mechanical Turk*. In his task, workers were given prompt words and were asked various sentiment-based questions, such as how positive or negative the word was and how much the word was associated to certain emotions, like *fear* and *trust*. As a result, the author obtained the sentiment polarity Ground-truth for 10170 sentiment words. To validate the lexicon, the author compared a subset of it with existing gold standard data, concluding that the obtained annotations were highly reliable.

In another study, Snow et al. used Crowdsourcing to obtain annotations for five different tasks [SOJN08]. In the first task, workers were asked to rate short headlines for six emotions, in a numeric interval from 0 to 100: 0 meant the specific emotion was not present in the headline while 100 meant the emotion was abundantly present. In a second task, workers were asked to rate the similarity between pairs of words in a similar fashion, in a scale from 0 to 10. The other three tasks consisted on recognizing textual entailments, event annotation and sentence disambiguation. *Amazon Mechanical Turk* was used and a total of 21690 judgements were obtained. The author aimed at using the obtained labels to study the reliability of Crowdsourcing workers when compared to experts.

In [DS10], Diakopoulos et al. used Crowdsourcing to characterize debate performances via Twitter. In his study, Diakopoulos captured tweets shared simultaneously with the first U.S. presidential debate in 2008 and posteriorly asked workers to evaluate the polarity of collected tweets. Then, by analysing the Ground-truth obtained for the tweets, the author was capable of understanding which of the two presidents was performing better at certain times of the debate. The corpus consisted of 3238 tweets and the Ground-truth was obtained with *Amazon Mechanical Turk*.

In the same year, Brew et al. [BGC10] presented a study where Crowdsourcing was used to obtain the polarity Ground-truth for online journal articles, obtained from *RTE*, *The Irish Times* and *The Irish Independent*. In his task, workers were asked to rate articles as *positive*, *negative* or *irrelevant*: the obtained annotations were then used to associate different topics to different sentiment polarities. The obtained Ground-truth was then used to train a machine learning algorithm to track sentiment in Online Media autonomously.

Finin et al. [FMKKMD10], in turn, presented a study where Crowdsourcing was used for annotating named Entities in Twitter data, via both *Amazon Mechanical Turk* and *Crowdfunder*. In his task, tweets were presented as data and workers were asked to specify, for each word of the tweet, if the word represented a Person, a Place, an Organization

or None. The dataset considered 506 different words from 30 different tweets and a total of 986 judgements were obtained. The author aimed at using the obtained data as Ground-truth to futurely train a framework to recognize named entities.

2.7 Summary

This chapter discussed previous work and background information relevant to the development of this dissertation. In the context of **Social-Media**, interesting work included:

- A study presented by Chen et al. [CWNWS12] where sentiment analysis was used to identify the sentimental polarity and domain corpus of unlabelled tweets. Additionally, a study presented by Ozdakis et al. where the usage of *hashtags* to improve event detection in tweets is tested and verified [OSO12].
- A study presented by Snow et al. where Crowdsourcing is proven to be a reliable medium to obtaining Ground-truth [SOJN08]. In a practical side: a study presented by Mohammad et al. where Crowdsourcing was used to generate a sentiment lexicon via *Amazon Mechanical Turk* [MT12]; a study presented by Brew et al. where Crowdsourcing was used to obtain the sentiment polarity of online articles [BGC10].

Likewise, in the context of **Recommendation Systems**, interesting work included:

- An hybrid recommendation approach proposed by Pazzani et al. [Paz99] where state-of-the-art Collaborative Filtering is enhanced by including Content-based user *profiles* and using those *profiles* to obtain the most similar users. The approach is proven to tackle *data sparsity*. In the movies context, an hybrid approach presented by Melville et al. where the rating matrix is enhanced with a Content-based predictor, improving recommendation when compared to pure state-of-the-art methods and linear combinations of both [MMN02].
- An hybrid movie recommender proposed by Moshfeghi et al. where sentiment analysis is used to extract emotion from textual reviews and plot summaries, in order to tackle *data sparsity* and the Cold-start problem [MPJ11]. Similarly, an hybrid method proposed by Zhang et al. where sentiment polarity is obtained from reviews and comments to improve on the problem of *data sparsity* for an online video service [ZDCL10].
- An alternative Collaborative Filtering approach proposed by Amatriain et al. where "expert" opinions are obtain from an external source to improve *data sparsity* and recommendation quality on a movie recommender [ALPKO09].



Measuring Reputations and Popularities in Social-Media

This chapter explains how the Social-Media feedback about new movies, directors and actors is captured and processed. First, a method to compute the reputation of directors and actors on IMDb is described [PSM14a]. Next, it is specified how tweets about new movies are acquired and classified. By the end of the chapter, ground-truth obtained via Crowdsourcing is used to validate both methods.

3.1 Introduction

Since their emergence, Social-Media platforms have been a preferred way to analyse the popularity and reputation of various brands, organizations and products [OBTR12; MWGA13; SAMAGG13]. These platforms have become such an esteemed source of feedback as a result of their accessibility, which propelled their increasing worldwide popularity. By collecting huge amount of information on subjects such as movies, books and other product, their importance became clear for recommender systems, whose goal is to compute suggestions concerning products of interest.

In this dissertation, Social-Media platforms are explored to obtain feedback on new movies and their respective directors and actors, in order to tackle the *cold-start* problem when recommending new movies. The popularity of new movies, which *have just been released*, is an immediate information that can only be captured on the present time: here, Twitter is monitored in order to capture the real-time popularity of new movies. For this purpose, collected tweets are also classified according to the positive or negative sentiment expressed towards the identified new movie. Contrary to the popularity of

new movies, the reputation of its directors and actors is not immediate but, instead, built over time. Hence, past movie reviews shared on IMDb are crawled in order to determine the reputation of the mentioned directors and actors: an external reputation framework [PSM14a] is used for this task, implemented by the primary author of the corresponding article. Figure 3.1 relates the reputation of new movies, their directors and actors to a simplistic timeline, in order to illustrate how these are built.

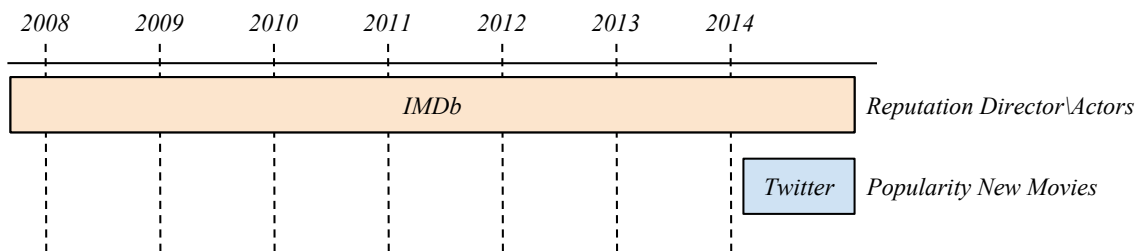


Figure 3.1: Timeline for the reputation of new movies, directors and actors.

By the end of the chapter, the monitoring process for both the popularity of new movies on Twitter and the reputation directors and actors on IMDb is validated by exploring two ground-truth datasets, collected by leveraging on Crowdsourcing practices.

3.2 Learning the Reputation of Entities

In IMDb, users are both able and encouraged to write textual reviews providing feedback on movies. These reviews commonly dwell into specific aspects of the target movie, providing insightful feedback on its various components. Directors and actors, who are highly relevant selling points for a movie, are generally targeted and analysed in these reviews. Considering this, it is obvious that written movie reviews can be a great asset when trying to estimate the reputation of these entities.

Here, a framework [PSM14a] is used to compute the reputation of directors and actors, given a set of IMDb reviews. Before introducing the framework, let us first consider the following sentence, extracted from a real review regarding the movie *Extremely Loud & Incredibly Close*:

*"Very moving and incredibly well acted by **Thomas Horn**, he completely outshines **Tom Hanks** and **Sandra Bullock** although their performances were also good."*

On a traditional sentiment analysis approach [PLV02; PL05; PL08], the reputation of the actor *Thomas Horn* can be estimated by considering sentiment words such as the unigram *incredibly* and the bigram *well acted*, which are expressing positive sentiment towards him. However, by analysing the sentence a little deeper, one can apprehend that the sentiment expressed towards both *Tom Hanks* and *Sandra Bullock* is also influencing the reputation of *Thomas Horn*: these entities are being used as comparative references to elevate *Thomas Horn*.

The considered framework builds on the idea that the reputation of an entity can be estimated not only by considering the sentiment words used towards it, but also the reputation of related entities, i.e. entities who are often compared or cross-referenced with the targeted entity. More specifically, the framework computes the reputation of named entities in a 3-steps process. First, the sentiment of each individual word used in the reviews corpus is determined, in order to build a domain-specific sentiment lexicon (Section 3.2.1). Secondly, both the corpus entities (i.e. the directors and actors) and their respective relations are identified in order to build a sentiment graph, linking the various identified entities (Section 3.2.2). Finally, the sentiment graph is used to iteratively compute the reputation of the named entities, by considering the reputations of adjacent entities in addition to the traditional computation based on the sentiment words (Section 3.2.3). In the context of this dissertation, the named entities are the directors and actors of the new movies. We can formalize the input of the framework as the set D of movie reviews,

$$D = \{(w_1, s_1), \dots, (w_i, s_i), \dots, (w_l, s_l)\}, \quad (3.1)$$

where a review tuple (w_i, s_i) contains the textual review in the form of a word vector $w_i = (w_{i1}, \dots, w_{iN})$ and an associated numeric rating $s_i \in \{1, \dots, 10\}$, where 1 corresponds to the worst rating and 10 corresponds to the best rating. The following subsections will address each step of the framework separately.

3.2.1 Building a Domain-Specific Sentiment Lexicon

The first step of the reputation analysis framework [PSM14a] is to build a domain-specific sentiment lexicon, containing the sentiment words used in the movie reviews corpus and the respective sentiment polarities. This lexicon is used when computing the reputation of entities in Section 3.2.3: popular sentiment lexicons [ES06] are too universal and, as a consequence, fail to capture relevant sentiment words that are often used in the movies domain. An obvious example is the word *Oscar*: while in a general context it is simply the name of a person, in the movies context it often refers to an award given to great movies, directors and actors.

To build the domain-specific lexicon, an alternative approach to The Latent Dirichlet Allocation (LDA) [BNJ03] is applied to the set of IMDb movie reviews. The LDA is, in short, a generative model that is capable of associating words to detected hidden topics by exploring word occurrences on documents. Furthermore, each word can be associated to various topics with different probabilities. When building the lexicon, the goal is to find words associated not to hidden topics, but to positive or negative sentiments. Hence, the LDA model is applied to the reviews of each numeric rating separately, in order to calculate and compare the probability of each word occurring on different ratings. The author refers to this approach as the Rank-LDA. Let $p(sw_i | s = 1)$ be the probability of a word w_i occurring on reviews of rating 1. Likewise, let $p(sw_i | s = 10)$ be the probability

of a word w_i occurring on reviews of rating 10. A sentiment weight for the word w_i is obtained by the expression

$$RLDA(w_i) = \frac{p(sw_i|s=10) - p(sw_i|s=1)}{\min(p(sw_i|s=10), p(sw_i|s=1))}, \quad (3.2)$$

concerning the variance of the word relevance for the best rating 10 and the worst rating 1. The resulting value will be a weight expressing the positivity or negativity level of the sentiment word sw_i on the lexicon, i.e. the lexicon contains multi-level polarity. The probabilities $p(sw_i|s=1)$ and $p(sw_i|s=10)$ are obtained from the built Rank-LDA model for each rating. Figure 3.2 illustrates the Rank-LDA graphical model.

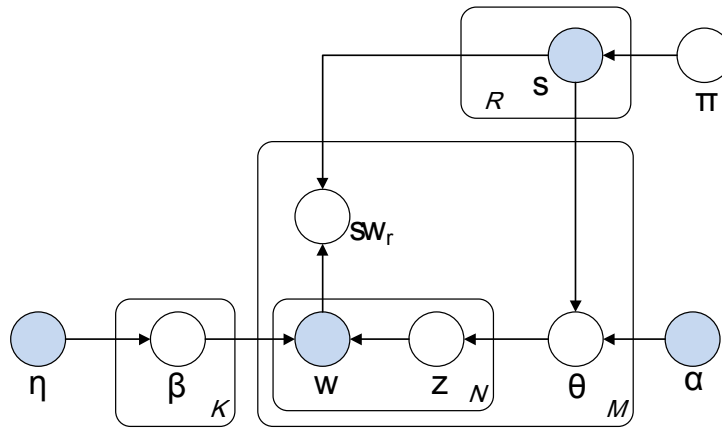


Figure 3.2: The Rank-LDA graphical model (from [PSM14a]).

In the Rank-LDA graphical model, R is a set of numeric ratings, where each rating $s_i \in \{1, \dots, 10\}$, K is the set of latent hidden topics, M is the set of reviews and N is the set of words in each review of M . The core of the model is as follows: for each word w , associate a set of detected hidden topics (where z is one associated topic) for each sentiment rating s . As a result, sw_r is the per-word sentiment distribution across the different ratings, i.e. contains the probabilities of a word w_r occurring for each different rating in R . In turn, β is the per-corpus topic $Dirichlet(\cdot|\eta)$ distribution, θ is the per-review topic $Dirichlet(\cdot|\alpha)$ distribution and α , η and π are random prior distribution variables (refer to [PSM14a] for details).

The relevance value $p(sw_i|s)$ of a word w_i to a rating s is then obtained by the sum of its probability for all latent topics identified for the rating s ,

$$p(sw_i|s) = \int p(\theta) \cdot \prod_{n=1}^N p(z_n|\theta, s) \cdot p(w_n|z_n) \cdot d\theta + \tau, \quad (3.3)$$

where $p(z_n|\theta, s)$ is the probability of the topic z_n occurring on the rating s , $p(w_n|z_n)$ is the probability of the word occurring on the topic z_n and τ is a smoothing parameter to avoid null values, set to 0.01.

3.2.2 Building a Linked-Entities Sentiment Graph

The reputation framework [PSM14a] is built on the idea that the reputation of an entity is measured by considering two different spaces: the reputation of related entities and the sentiment words used towards the target entity. Hence, before calculating the reputation of directors and actors, a sentiment graph is built to describe the relations between them and other entities of the corpus.

The sentiment graph construction process starts by identifying all the entities mentioned in the given IMDb reviews. To do so, the method starts by extracting the metadata of all reviewed movies, namely the list of directors, actors and characters, as all those are very likely to have been mentioned. Note that characters are also considered: the sentiment expressed towards a movie character is usually transitive to the respective actor. However, movie reviews often also mention entities that are not directly related to the reviewed movie. To capture these entities, the framework uses the NLTK Named-Entities and Relation extractor¹. This tool is capable of automatically discovering and extracting relevant entities on textual documents.

After the entities are extracted, two types of relations between them are characterized, namely the explicit and implicit relations. The explicit relations are obtained by identifying co-occurrences between entities in the same review sentences. These relations are formalized as

$$\psi(e_i, e_j) = \frac{\#(e_i, e_j)}{\#(e_i) + \#(e_j)}, \quad (3.4)$$

where $\#(e_i, e_j)$ is the number of times the entities e_i and e_j co-occur together and $\#(e_i)$ is the number of times an entity occurs individually. In turn, the implicit relations are obtained via the Rank-LDA model, described in Section 3.2.1. After building the model, an implicit relation between two entities is identified if they are associated to the same latent topic. Furthermore, since the model is applied to the reviews of the different ratings separately, relations at different sentiment levels are identified. The implicit relation $slda(e_i, e_j)$ between an entity e_i and an entity e_j is formalized as

$$slda(e_i, e_j) = \sum_{r \in R} \sum_{z \in Z} (p(e_i, z) + p(e_j, z)), \exists e_i, e_j \in z, \quad (3.5)$$

where R is the set of all ratings, Z is the set of all detected latent topics, $p(e_i, z)$ is the relevance of entity e_i for the latent topic z and $p(e_j, z)$ is the relevance of entity e_j for the same topic.

A sentiment graph is finally constructed for representing the sentiment relations not only between entities, but also between entities and sentiment words. Hence, the sentiment graph $ER = (V, R)$ is a single heterogeneous graph, where the set of vertices V

¹www.nltk.org

corresponds to the identified named-entities and the set of edges R represents the relations between entities and sentiment words. Figure 3.3 illustrates the structure of the sentiment graph ER , where $h(e_i, e_j)$ is a relation between two entities and $f(e_i, sw_j)$ is a relation between an entity and a sentiment word.

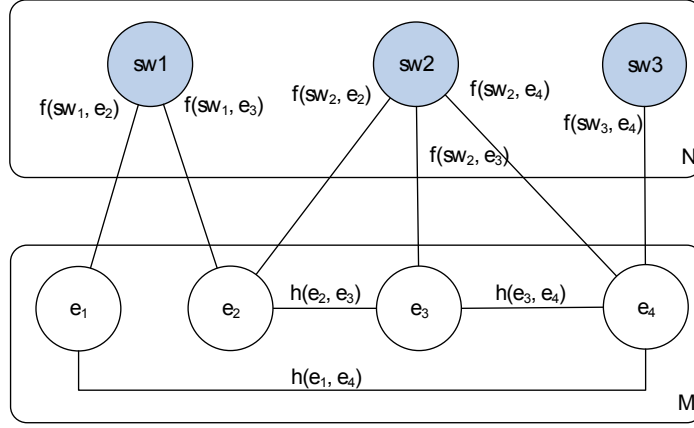


Figure 3.3: The linked-entities sentiment graph structure (from [PSM14a]).

The reputation of an entity is followingly calculated by leveraging on the sentiment strength of all its adjacent $h(e_i, e_k)$ and $f(e_i, sw_k)$ relations.

3.2.3 Computing the Reputation of Entities

After building the sentiment graph, all sentiment relations towards entities are specified: for an entity, we known both who are the related entities and what sentiment words describe the entity. Hence, its reputation can be calculated by considering the sentiment of all its adjacent relations. The sentiment strength of a sentiment relation $h(e_i, e_j)$ between an entity e_i and an entity e_j is obtained by the expression

$$h(e_i, e_j) = slda(e_i, e_j) + \frac{RLDA(e_i) + RLDA(e_j)}{2}, \quad (3.6)$$

where $slda(e_i, e_j)$ comprises the value of the implicit relation between the entities, obtained via Equation 3.5, and $RLDA(e_i)$ is the Rank-LDA model polarity weight attributed to the entity e_i , obtained via Equation 3.2. In turn, the sentiment strength of a relation $f(e_i, sw_n)$ between an entity e_i and a related sentiment word sw_n is obtained by the expression

$$f(e_i, sw_n) = \frac{RLDA(e_i) + RLDA(sw_n)}{2}, \quad (3.7)$$

where $RLDA(sw_n)$ is the sentiment weight associated to the sentiment word sw_n on the domain-specific sentiment lexicon, built in Section 3.2.1.

The reputation of an entity e_i can finally be calculated in light of Equations 3.6 and 3.7. First, the full sentiment expressed towards the entity e_i via the related sentiment words

is calculated. Hence, the sentiment value $f_0(e_i)$ expressed by sentiment words towards e_i is obtained by the equation

$$f_0(e_i) = \sum_{sw_n \in E \cup SW} f_n(e_i, sw_n), \quad (3.8)$$

where $E \cup SW$ is the set of sentiment words related to e_i . The final reputation $rep(e_i)$ of an entity e_i is then obtained by considering both $f_0(e_i)$ and the reputation of its adjacent entities,

$$rep(e_i) = f_0(e_i) + \sum_{e_j \in \{N(e_i)\}} \frac{rep(e_j)}{\# \{N(e_j)\}} \cdot \psi(e_i, e_j) \cdot h(e_i, e_j), \quad (3.9)$$

where $N(e_i)$ is the set of entities related to e_i , $rep(e_j)$ is the reputation of a related entity e_j , $\psi(e_i, e_j)$ is the value of the explicit relation between e_i and the related entity e_j (Equation 3.4) and $h(e_i, e_j)$ is the value of the implicit graph relation between e_i and the related entity e_j (Equation 3.6). Note that this is an iterative process: since the reputation of a related entity might be influenced by the reputation of the target entity, the reputations need to be revisited until their value stagnates. Algorithm 1 details the iterative reputation computation for all the corpus entities, initiated by receiving the sentiment graph ER , the polarity value of all sentiment words $RLDA(sw_*)$ and the polarity value of all corpus entities $RLDA(e_*)$.

Algorithm 1 Iterative Reputation RLDA Algorithm

Input: $ER, RLDA(sw_*), RLDA(e_*)$

Output: $rep \leftarrow$ Set of entities and respective reputations.

```

1: for all  $e_i \in E$  do
2:   for all  $e_j \in N(e_i)$  do
3:      $h(e_i, e_j) = slda(e_i, e_j) + (RLDA(e_i) + RLDA(e_j))/2$ 
4:      $\psi(e_i, e_j) = \#(e_i, e_j) / (\#(e_i) + \#(e_j))$ 
5:   end for
6: end for
7: repeat
8:   for all  $e_i \in E$  do
9:      $rep(e_i) = f_0(e_i)$ 
10:    for all  $e_j \in N(e_i)$  do
11:       $rep(e_i) += rep(e_j) \cdot \psi(e_i, e_j) \cdot h(e_i, e_j)$ 
12:    end for
13:  end for
14: until all  $rep_{i \rightarrow j}(e_j)$  and  $rep_{i \rightarrow n}(e_n)$  stop changing.
15: return  $rep$ 

```

The algorithm starts by computing values of all explicit ($h(e_i, e_j)$) and implicit ($\psi(e_i, e_j)$) entity-entity relations for all entities. After that, the reputation of each entity is updated until the reputations for all entities stabilize. The reputation of an entity is obtained, in

each update cycle, by considering both the sentiment value obtained by the sentiment words related to it, i.e. $f_0(e_i)$, and the strength of its relation to all its adjacent entities, $rep(e_j) \cdot \psi(e_i, e_j) \cdot h(e_i, e_j)$.

The resulting entities and respective reputations are indexed to the names of the new movies where they have participated, in order to allow look-ups by movie title. Formally, the reputation of all the directors and actors participating on a new movie m_j is represented by the expression

$$reps(m_j) = \{rep(e_1), \dots, rep(e_k), \dots\}, \quad (3.10)$$

where the reputation of each entity e_k is normalized so $rep(e_k) \in [0.0, 1.0]$, with 0.0 being the worst reputation and 1.0 being the best reputation.

3.3 Twitter Mining and Classification

As previously disclosed, Twitter is monitored in order to capture feedback about new movies: Twitter users share small textual posts concerning various subjects, including cinematography, from which the popularity of new movies can be calculated. The Twitter mining process starts with the capture of user-shared tweets by using the official public Twitter streaming API². This API enables developers to capture a sample of real-time shared tweets, consisting of 10 Million tweets per day. According to Mathioudakis et al. [MK10], the estimated total of tweets shared per day is 50 Million, meaning that a fifth of all the shared tweets is captured. Each collected tweet also contains various additional information, such as its id, a timestamp, the id of the user who shared it and its language. Formally, a captured tweet t_k is represented as the vector

$$t_k = (t_{txt}, t_{id}, u_{id}, t_{time}, t_{lang}), \quad (3.11)$$

where t_{txt} is the text in string format, t_{id} is its id, u_{id} is the id of the user who shared it, t_{time} is the timestamp and t_{lang} is its language. From this information, the tweets that are not written in English can be immediately discarded.

While the mining process is performed, the tweets that are targeting new movies are identified. A tweet is considered to be targeting a new movie if one of the following two conditions is verified: first, if its text contains the title of the movie; second, if its text contains an *hashtag* referring the movie. Twitter *hashtags* are represented by starting with an # character and are usually deprived of white spaces, e.g. the *hashtag* "#IronMan3" refers the movie "Iron Man 3". Figure 3.4 shows a realistic example of two tweets identified to be targeting the movie "Iron Man 3" by containing the movie title and an *hashtag* referring it, respectively.

Each tweet is tested against all the considered new movies and is thereby tagged according to the identified movies, i.e. tweets referring various new movies are tagged

²dev.twitter.com/streaming/public

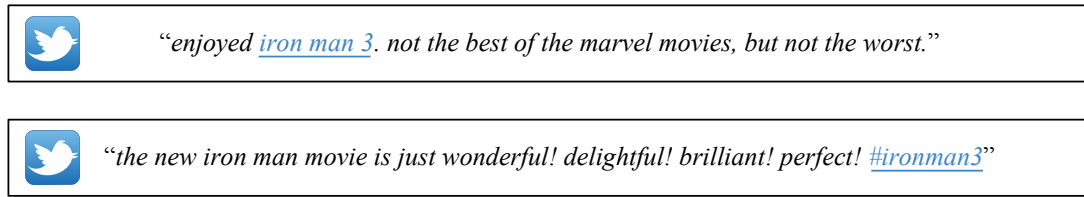


Figure 3.4: Example of tweets identified for the new movie "Iron Man 3".

with the names of all referred movies. As a result, only the tweets concerning at least one new movie are considered relevant and stored.

After the mining process is concluded, each resulting tweet is classified in order to infer its sentiment polarity, i.e. if it contains positive or negative sentiment towards the referred movies. The k -NN algorithm [ESK03] is used for this classification task, as it is one of the simplest and overall best performing classification algorithms [FS07]. In k -NN, each data sample is represented as a set of features and the corresponding value for each feature. The most common approach in textual classification tasks is to use the sentiment words present on the text as the features [PLV02; PL05; PL08]. Hence, each tweet is pre-processed in order to extract its sentiment words and the respective values.

The first pre-processing step is to remove the names of the identified movies from the tweet text, since the sentiment expressed towards them is usually conveyed on the rest of the sentence. This step reduces the noise during the classification task, e.g. the word *great* in the movie title "The Great Gatsby" would be regarded as a sentiment word and influence the classification of tweets referring that movie. After removing the title of the movies, the rest of the sentence is split by white spaces, resulting in a vector where each element is a word or a punctuation symbol. A third step removes both the punctuation symbols and the stop words from the vector: stopwords are words that carry no sentiment and are therefore irrelevant for sentiment analysis tasks, such as the words *as*, *at*, *is* and *the*. Removing punctuation and stop words improves the performance of the classification task, as it discards unnecessary features. While the resulting vector already contains only the sentiment words, these are converted to their canonical form, e.g. the words *amazing* and *amazement* are converted to *amaze*, as these present the same sentiment and are therefore regarded as the same sentiment word. A final step uses the domain-specific sentiment lexicon built in 3.2.1 to associate the sentiment values to the found sentiment words. As a simplistic example, the tweet "I love Iron Man 3, it is the best movie ever!" translates into the feature vector "[*(love, 0.7543)*, *(best, 0.9026)*, *(movie, 0.6427)*, *(ever, 0.6402)*]". Figure 3.5 illustrated the feature extraction process described in this paragraph.

A total of 300 tweets are used as the test set for k -NN, where 150 are labelled as positive and 150 are labelled as negative: the labels were obtained via Crowdsourcing (Section 3.4). A tweet is classified by considering the labels of the $k = 10$ most similar training

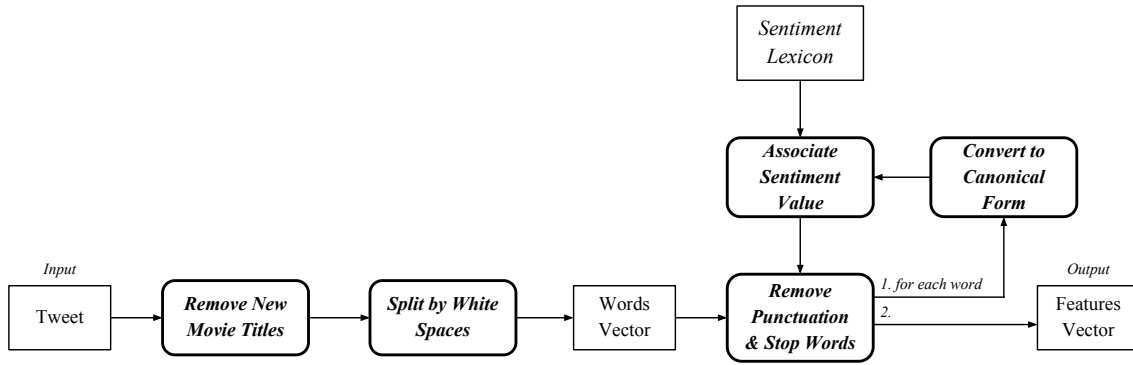


Figure 3.5: Feature extraction process for tweets referring new movies.

tweets to the target tweet: the tweet is *positive* if 7 of the most similar training tweets are positive, *negative* if 7 of the most similar are negative and *neutral* otherwise. The similarity of the tweets is computed by applying the Manhattan distance to their corresponding feature vectors.

The classified tweets are finally indexed by the corresponding new movie tags to allow fast look-ups, given the title of a new movie. Tweets classified as *neutral* are not indexed, as these are generally associated to spam or objective sentences, where sentiment is not expressed. Formally, the set of classified tweet $T(m_j)$ referring the new movie m_j , obtained by the Twitter monitoring process, is then defined as the set

$$T(m_j) = \{(t_{j1}, s_{j1}), \dots, (t_{jl}, s_{jl}), \dots, (t_{jM}, s_{jM})\}, \quad (3.12)$$

where t_{jl} is the textual tweet (referring m_j) and s_{jl} is the corresponding sentiment polarity such that $s_{jl} \in \{pos, neg\}$.

3.4 Crowdsourcing for Social-Media Ground-Truth

In order to validate the media monitoring methods presented in Sections 3.2 and 3.3, the appropriated ground-truth is needed. Crowdsourcing is used to obtain this data³ as it is a novel, reliable and cheap way to obtain ground-truth [YKL11a]. Furthermore, several studies have successfully used Crowdsourcing for classifying tweets and other textual data [SOJN08; DS10; BGC10; FMKKMD10]. Two sets of ground-truth were obtained in order to validate the presented methods:

- **Ground-truth for IMDb sentences:** The first social-media monitoring component computes the reputation of directors and actors from IMDb review sentences. Therefore, the obtained ground-truth describes if a sentence is truly referring the directors or actor positively or negatively. The accuracy of the reputation algorithm is

³www.crowdfunder.com

validated by comparing the inferred reputations to the overall sentiment polarity expressed in the ground-truth sentences.

- **Ground-truth for tweets:** The second social-media monitoring component captures tweets regarding new movies and classifies their sentiment polarity towards the movie. Hence, the obtained ground-truth describes if the tweet is truly referring the movie positively, negatively or neutrally. The accuracy of the tweets classifier is validated by comparing the inferred labels to the ground-truth labels.

After identifying the desired ground-truth, the Crowdsourcing tasks were carefully designed in order to obtain the said data in the most reliable and resource-friendly manner possible. To do so, each component of the task was defined separately in a sequential manner. Sections 3.4.1 to 3.4.4 describe the designing process of each task component in order of conception, while Section 3.4.5 uses the resulting ground-truth to validate the proposed media monitoring methods.

3.4.1 Worker Interfaces

The first component to be defined for both tasks was the Worker Interface, which specifies how workers interact with the data. In a Crowdsourcing task, this interaction is generally performed by the workers answering to questions specified by the requester. The main concerns taken into account when defining the questions were:

- The answers to the questions provide the necessary information to later validate the methods;
- The questions are clear and easy to understand by the worker;
- Answering to the questions takes as less time as possible.

To collect the ground-truth for the IMDb sentences regarding entities, an interface was defined where the worker was asked to label a sentence in accordance to the expressed sentiment towards a specified named entity. In this interface, the worker could select one of four labels to describe the sentiment: *very positive*, *positive*, *negative* or *very negative*. Textual labels were selected since, according to Friedman et al. [FA99], rating something in words is the easiest approach, since that is how people express opinions everyday. Furthermore, the task was designed as a multi-level task since the reputation framework [PSM14a] obtains a multi-level reputation for entities, i.e. we want a similar level of sentiment complexity expressed in the ground-truth. For this task, only sentences where at last one sentiment word and one entity were identified were considered: as a consequence, workers did not have to analyse if the sentence was sentimentally neutral. Figure 3.6 illustrates the worker interface for the described task.

In turn, to collect the ground-truth for the tweets regarding new movies, an interface was built where the worker was asked to label a tweet according to the sentiment expressed towards the movie, as either *positive*, *negative* or *neutral*. Differently to the first

Consider the sentence, extracted from a movie review:

"The actors, Bruce Willis and Samuel Jackson, were okay, but the story is horrible!"

In the sentence - Bruce Willis - reputation is positive or negative?

- ☐ Very Positive
- ☐ Positive
- ☐ Negative
- ☐ Very Negative

Figure 3.6: Worker interface for the IMDb sentences task.

task, this task was designed as a binary relevance task, since that is how the k -NN classifier classifies the various tweets. The workers were also asked to classify neutral tweets, since our method identifies and filters neutral tweets. Figure 3.7 illustrates the worker interface for the tweets task.

Consider the tweet, extracted during the movie oscars 2014:

"finished the book thief... wow is all i can say... bloody fantastic"

How is the tweet referring the movie "The Book Thief"?

- ☐ Positively
- ☐ Negatively
- ☐ Neutrally


 hint: select "neutrally" when it is not speaking positively or negatively about the respective movie.

Figure 3.7: Worker interface for the tweets task.

To assess the efficacy of the designed interfaces, a trial run was performed for each task, using 50 IMDb sentences and 50 tweets. After performing a Crowdsourcing task, a worker can rate the satisfaction towards the performed task in various fields, in a rating scale from 1 to 5: here, we take into account the *Instructions Clear* and *Ease of Job* fields. The IMDb task was rated by 41 workers, obtaining 4.2 in terms of instructions and 4 in terms of easiness. The tweets task, in turn, was rated by 44 workers, obtaining 4.6 and 4.2 for the same fields, respectively.

3.4.2 Worker Qualification

After modelling the Worker's Interface, the qualification criteria for Workers to be accepted in the defined tasks was specified. Applying the right selection criteria for a Crowdsourcing task is important, since some workers might not be qualified to execute a specific job and their participation might negatively influence the reliability of the obtained results. In the case of the required tasks, no special or professional skill is needed. Therefore, a worker is accepted to perform any of the tasks depending on the following

criteria:

- **Geography:** Both datasets are highly related to Hollywood cinematography, as they target mostly American movies and entities. Some countries, like India, have their own cinematography culture, resulting in different quality standards. Additionally, the datasets only include reviews and tweets written in English. To ensure that all workers are compatible with the datasets, only workers from countries where English is the main language were accepted, namely Australia, Canada, United Kingdom and United States.
- **Test Questions:** Test questions are questions where only some answers are accepted and are used to filter workers with poor performance. Generally, only questions where the answer is obvious or almost obvious are selected as test questions. In the defined tasks, a worker needed to answer correctly to a certain number of test questions before being able to answer the normal questions. In addition, hidden test questions that are mixed with the normal ones were used in order to exclude workers that stopped putting effort halfway.

3.4.3 Tasks Parameters

In addition to Worker's Qualification, there are other task attributes that directly influence the reliability of the obtained results. Task Parameters are closely related to how the task proceeds while running and not only influence the obtained results, but the job's cost as well. Before running the main tasks, various trial tasks with small subsets of the datasets were executed so the best parameters could be estimated. The estimated parameters were the following:

- **Price per Page:** Crowdsourcing tasks are generally presented in pages and a worker decides how many pages to complete. This parameter defines how much the Worker is paid per completed Page. The higher the payment, the more motivated the worker is, so it's important to estimate how much is enough for the worker to put enough effort on the job.
- **Units per Page:** For each task, the defined question is repeated for each unit of the dataset. This parameter defines the number of different units presented per Page. Small numbers of units might motivate the worker to contribute more, since it doesn't force him to work too much for each payment. However, this increases the cost per unit, so it's important to find an appropriated middle term.
- **Judgements per Unit:** How many Workers judgements are collected per Unit. The more judgements collected, the more reliable the obtained results are. However, the overall cost also increases, so it's important to define how many judgements are enough to obtain solid results.

A total of 6 trial tasks with different parameters were executed: 3 for each task of the presented in 3.4.1, i.e. 3 for the tweets classification task and 3 for the IMDb sentences classification task. In order to reduce the number of trial tests and total costs of the trial phase, the number of units per page was not directly estimated and was always designated as 10: in the first trial task it was observed that most workers completed more than 10 units, averaging 39.23 units. To rate the obtained results, two result parameters were considered:

- **Agreement:** The agreement of a unit describes how similar the obtained labels for that unit were. Although its natural that some units are harder to label and that the agreement for those units can be low, if a workers put enough effort the overall agreement of the dataset is usually medium-high. A very low agreement generally means either that the task is too complex or that the workers are not motivated enough.
- **User Satisfaction:** A worker can rate the satisfaction level of an executed task in various fields, in a scale from 1 to 5. In this case, the *Payment Satisfaction* is considered as a representation of workers motivation. With higher motivation, the probability of the results being reliable is also higher.

Datasets of approximately 100 units were used in all trial runs. The tested parameters and obtained results are presented in Table 3.1.

Task	Cost/Page	Judg./Unit	Agreement	Satisfaction
IMDb	0.01\$	3	68.04%	3
IMDb	0.02\$	3	64.87%	3.9
IMDb	0.02\$	5	65.08%	4.2
Twitter	0.01\$	3	82.12%	3.8
Twitter	0.02\$	3	80.01%	4.0
Twitter	0.01\$	5	80.98%	3.7

Table 3.1: Trial tasks parameters and results.

Hsueh et al. presented a study [HMS09] where Crowdsourcing was used to label textual blog snippets, in which a sentiment polarity task (*positive* vs *neutral* vs *negative*) obtained an overall agreement of 61.9%. In comparison, the obtained results were very good: the IMDb task presented slightly better agreement results despite being more complex (it is a *multi-level* task); in turn, the Twitter task presented much better results in a similar task approach. However, in the IMDb task most users that expressed their satisfaction thought that 0.01\$ was not enough payment, while a payment of 0.02\$ was agreed to be enough. The same is not observed for the Twitter task, probably due to the easier nature of that task. This shows that the Cost parameter does not need to be increased any further for any of the tasks. Furthermore, the number of judgements per unit does

not seem to have any direct influence on the agreement or satisfaction results: we can set this parameter in accordance to the possessed funds, but 3 judgements seem to be acceptable. The slight fluctuation on the overall agreement values is to be expected, since most workers are different on all runs and some runs might end up having better workers than others, obtaining better results.

3.4.4 Tasks Execution

After modelling the Worker Interface (3.4.1), specifying Worker Qualification (3.4.2) and estimating Task Parameters (3.4.3), the main tasks were launched to collect the desired labels. For the IMDb sentences sentiment task, a total of 4,000 sentences regarding at least one of 20 popular manually selected entities were submitted. From those, roughly 1,500 referred more than one entity. For the tweets classification task, a total of 4,000 tweets regarding at least one of 60 manually selected new movies were submitted (refer to Section 5.1 for more information on the selected new movies). From those, roughly 1,000 were identified by *hashtag*. Snow et al. [SOJN08] showed that an average of 4 non-expert workers are able to match the quality of expert annotators, so judgements from 5 different workers were collected for each sentence and tweet, in order to best emulate expert labelling. Furthermore, 20 units were used in each task as test units, in order to filter untrustworthy workers. Table 3.2 summarizes the selected parameters for the launched tasks.

Task	Units	Test Units	Units/Page	Judg./Unit	Cost/Page
IMDb	4,000	20	10	5	0.02\$
Twitter	4,000	20	10	5	0.01\$

Table 3.2: Main tasks parameters.

In the end, a total of approximately 20,000 judgements were collected for each tasks, resulting in a total cost of 60\$. The entities reputation and tweets classification methods are followingly validated with the obtained ground-truth.

3.4.5 Results and Discussion

3.4.5.1 IMDb Reputation

Figure 3.8 summarizes the distribution of the obtained labels for the IMDb sentences task, comprising of 4,000 sentences referring 20 different entities. By analysing 3.8a, it can be observed that most of the sentences were labelled with an agreement bellow 70%, with the overall average agreement being 73%. This shows that the sentiment expressed toward entities in most sentences is not trivial to judge accurately. In turn, 3.8b shows the labels frequency per agreement value. The most frequent labels to obtain low agreements were the *positive* and *negative* labels, while the most frequent labels to obtain an agreement

of 100% were *very positive* and *very negative*, i.e. sentences expressing a strong sentiment towards entities are easier to identify and classify. Furthermore, 77.36% of the sentences were labelled as *positive* or *very positive*, while the remaining 22.64% were labelled as *negative* or *very negative*.

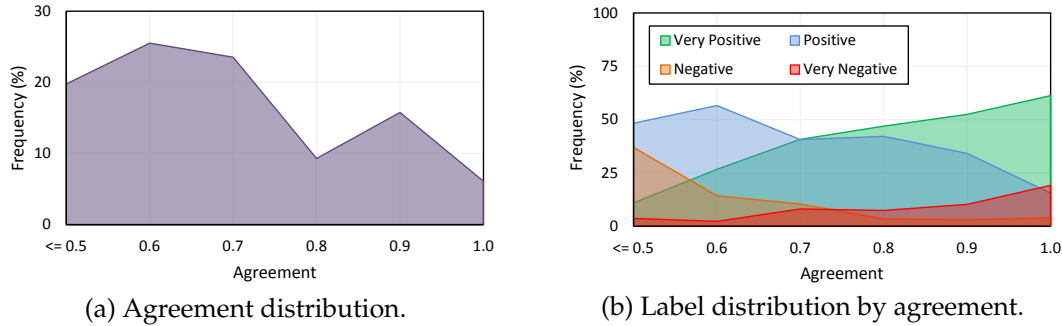


Figure 3.8: Distribution of Crowdsourcing labels for IMDb sentences.

To validate the Reputation Analysis method [PSM14a] presented in Section 3.2, the reputation of 8 popular directors and actors were computed by using the obtained ground-truth: 200 sentences, evenly distributed by label, were used as the training set, while the rest was used as the test set. Since the Ground-Truth does not concern the reputation of the entities in a numeric scale, like the output of the algorithm, we compare how the algorithm labels each test set sentence to its respective Ground-Truth label. Table 3.3 shows the obtained accuracy for the 8 selected entities.

Entity	Accuracy (%)
Bruce Willis	87.50
Colin Firth	84.21
Johnny Depp	96.25
Miley Cyrus	88.89
Peter Jackson	87.80
Shia Labeouf	78.57
Stanley Kubrick	94.44
Woody Allen	94.44
Average	89.01

Table 3.3: Reputation analysis accuracy for 8 popular directors and actors (%).

Overall, the method obtained an average accuracy of 89.01%, which is extremely high and validates the method for the desired context. Furthermore, even the weakest performing entity, namely *Shia Labeouf*, has obtained a classification accuracy of 78.57%, enforcing the conclusion that the method is robust for classifying the sentiment on IMDb sentences towards directors and actors. For a comparison of the obtained results with various popular baselines methods, refer to [PSM14a].

3.4.5.2 Twitter Classification

Figure 3.9 summarizes the distribution of the obtained labels for the tweets classification task, comprising 4,000 tweets referring 60 distinct movies. By analysing 3.9a, it can be observed that most sentences were labelled with 100% agreement, totalling 36.3% of the tweets dataset. Furthermore, the overall obtained agreement was 82.7%, suggesting that the sentiment expressed on tweets regarding new movies is relatively easy to describe. In turn, 3.9b shows the labels frequency by agreement value. The most frequent label to obtain low agreement values was *neutral*, suggesting that neutral tweets are the overall hardest to identify, when compared to tweets expressing positive or negative sentiment. While this value starts decreasing for higher agreement values, a great amount of neutral tweets have also obtained 100% agreement: these are most likely the tweets containing *spam*, which are easily identifiable by the workers. Overall, 54.1% of the tweets were labelled as *positive*, 14.1% were labelled as *negative* and the remaining 31.8% were labelled as *neutral*.

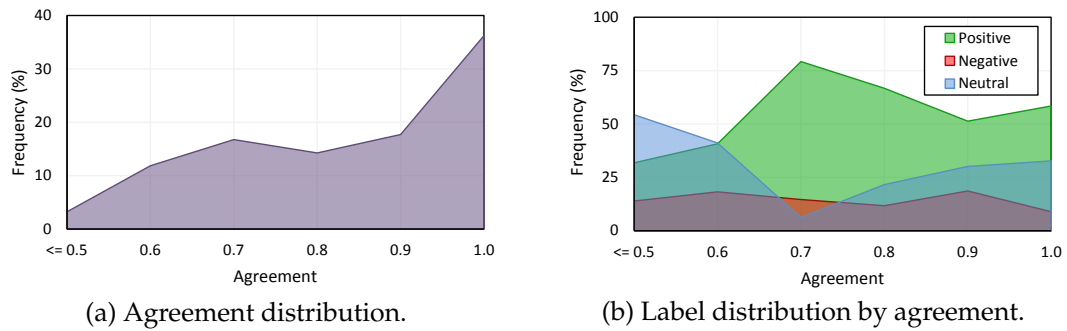


Figure 3.9: Distribution of Crowdsourcing labels for tweets.

To assess the accuracy of the tweets classification approach presented in Section 3.3, 300 tweets were used as the training set to classify the remaining tweets, i.e. these were the test set. Figure 3.10 plots the classification accuracy curves for the tweets of the various agreement levels in two variants: the *Unfiltered* curve presents the accuracy for all the test tweets, while the *Filtered* curve presents the accuracy for the tweets that were classified as *positive* or *negative* by the proposed method, i.e. the *Filtered* curve is the best representative of the proposed method, as it filters tweets classified as *neutral*.

The best results were obtained for tweets with higher agreement values, reaching a maximum accuracy of 81.8% for the *Unfiltered* variant and 85.2% for the *Filtered* variant. These results suggest that tweets that are easier to label manually are also easier to classify by the proposed method. In addition, there also seems to be an inverse relation between the frequency of *neutral* tweets and resulting accuracy: the best results are obtained when the frequency of neutral tweets is lower (agreement of 0.7), while the worst results are obtained when neutral tweets are more frequent (agreement ≤ 0.5). Overall, filtering the neutral tweets improves the classification accuracy, with the *Filtered* variant obtaining an average accuracy of 79.3% while the *Unfiltered* variant obtains an average accuracy of

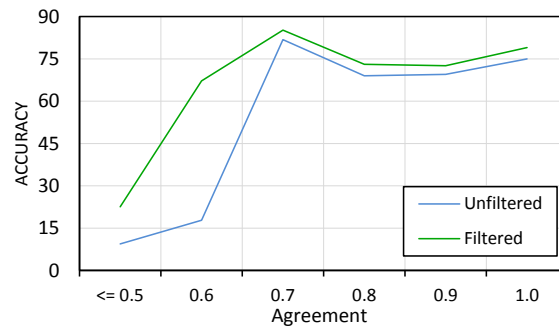


Figure 3.10: Twitter classification accuracy by agreement.

65.9%. This, together with the fact that better results are obtained when neutral tweets are scarcer, implies that the proposed method does not handle low levels of sentiment as well as extreme levels of sentiment, where the polarity of the tweet is more evident.

3.5 Summary

This chapter presented and validated the social-media monitoring methods to capture the popularity of new movies on Twitter and the reputation of its directors and actors on IMDb. The most important work presented on this chapter included:

- The reputation of a director or actor is calculated with a method [PSM14a] that explores the sentiment on IMDb reviews where those entities are mentioned. The method also builds a sentiment lexicon specific to the movies domain;
- The tweets regarding new movies are identified by the occurrence of their title on the tweet or by *hashtag*. A k -NN [ESK03] classifier is used together with the movie-specific sentiment lexicon created by [PSM14a] to classify tweets as *positive* or *negative* and filter neutral or unrelated tweets;
- Crowdsourcing is used to obtain ground-truth for IMDb review sentences and tweets, in order to validate both media monitoring methods. A total of 60\$ is spent to obtain the ground-truth for 4,000 IMDb sentences and 4,000 tweets.

4

Cold-Start Recommendations

This chapter details the implemented algorithm for recommending *cold-start* movies, by exploring Social-Media trends and reputations. First, the recommendation scenario is introduced and the main problem is formalized. The various components of the formal recommendation model are then detailed separately. By the end of the chapter, Social-Media signals are used to model recommendations.

4.1 Introduction

Nowadays, popular recommendation methods explore user-product feedback, such as numeric ratings, to relate different users or products and predict what products specific users would like to consume. These approaches are, however, susceptible to the *cold-start* problem [AT05]: new products that have not yet been rated cannot be related to other products or users and, consequently, will not be recommended. In this dissertation, the goal is to tackle the *cold-start* problem for new movies that have not been rated. Moshfeghi et al. [MPJ11] showed that recommendation of *cold-start* movies performs best when considering both the movie metadata and the sentiment expressed in its written reviews. Building on this idea, a movie m_j is here represented as the vector

$$m_j = (D_j, A_j, G_j, R_j, S_j), \quad (4.1)$$

where D_j is the set of directors, A_j is the set of the participating actors, G_j is the set of corresponding genres, R_j is the set of associated user ratings and S_j is the Social-Media feedback inferred by a monitoring process, described in Chapter 3. The sets D_j , A_j and G_j comprise the movie metadata. Some examples of features contained in these sets are the

names of popular directors and actors such as *Peter Jackson* and *Johnny Depp*, or common film genres such as *romance* and *horror*. In turn, the S_j variable is composed of the Twitter posts (or tweets) about the movie m_j as well as the reputation of its directors and actors, obtained from IMDb: these contain sentiment expressed toward the movie. As will be clarified further into the chapter, S_j will be fundamental to recommend *cold-start* new movies, where $R_j = \emptyset$.

Our recommendation scenario considers two main entities: a target user u_i and a set of movies M . In this scenario, users have rated the movies they have previously watched. Following on this information, a user u_i is described by the set of K ratings concerning the movies watched and rated by that particular user, formalized as the set

$$ur_i = \{r_{ih}^1, \dots, r_{ij}^k, \dots, r_{il}^K\}, \quad (4.2)$$

where each rating r_{ij}^k concerns the movie m_j . In turn, the set of movies M contains movies represented as in Equation 4.1. M is formalized as the set

$$M = \{m_1, \dots, m_v, \dots, m_V, \dots, m_c, \dots, m_{V+C}\}, \quad (4.3)$$

containing V rated movies and C unrated new movies, i.e. movies affected by the *cold-start* problem. In M , for any arbitrary viewed movie m_v , $R_v \neq \emptyset$, as these movies have been previously rated by users. Oppositely, for any arbitrary *cold-start* movie m_c , $R_c = \emptyset$, since new movie have not yet been rated.

Considering the previous formalizations, the goal of this chapter can be formally defined as follows: given an user u_i and a set of movies M , discover the set $R_i \subseteq \{m_{V+1}, \dots, m_c, \dots, m_{V+C}\}$ containing the new movies that user u_i would like to watch. More specifically, the set R_i comprises the *cold-start* movies for which it is predicted that the user u_i will rate above his user-specific quality threshold T^i . The objective set R_i is then formalized as the set

$$R_i = \{(m_h, \hat{r}_{ih})^1, \dots, (m_l, \hat{r}_{il})^p, \dots, (m_n, \hat{r}_{in})^P\}, \hat{r}_{il} \geq T^i, \quad (4.4)$$

where an arbitrary element $(m_l, \hat{r}_{il})^p$ consists of a recommended movie m_l and the respective predicted user-movie rating \hat{r}_{il} . This chapter focuses on explaining the rating prediction process, which solves Equation 4.4.

4.2 Building User Profiles

Before tackling the problem of predicting what *cold-start* movies to recommend to a user, a representation that clearly expresses user preferences is needed. While a set of previously given ratings is somewhat informative, by itself it does not provide sufficient information to predict how much that user will like a new movie. Thus, a profile is built to specify what characteristics the user likes and dislikes in movies. Remember the

representation of a movie, formalized in Equation 4.1: movies are represented by their metadata, namely a set of directors, a set of actors and a set of genres. Following on it, the profile of a user u_i is similarly formalized as the vector

$$u_i = (D^i, A^i, G^i, T^i, bias^i), \quad (4.5)$$

where D^i , A^i and G^i represent the user preferences towards director, actors and genres, respectively. Realistically, however, users are not only characterized by their preferences towards certain characteristics, but also by their personal standards: some users are easier to please than others and, consequently, different users have different rating patterns. Building on this fact, a user profile u_i also contains both the user-specific threshold, T^i , and the user-specific rating bias, $bias^i$.

4.2.1 Discovering User Preferences

In a user profile represented by Equation 4.5, the sets D^i , A^i and G^i comprise the user preferences towards directors, actors and genres, respectively. Since this information is not directly specified by the user, its discovery process leverages on a combination of the user previously given ratings and the metadata of the corresponding rated movies. In other words, how much a user likes a certain characteristic (e.g. a specific director) is estimated by analysing how the user has rated the movies containing that characteristic. Hence, the set D^i comprising the directors of the movies rated by u_i is formalized as the set

$$D^i = \{(d_i^1, dr_i^1, df_i^1), \dots, (d_i^n, dr_i^n, df_i^n), \dots\}, \quad (4.6)$$

where, for an entry (d_i^n, dr_i^n, df_i^n) , the first element d_i^n identifies the director, dr_i^n is the average rating given by the user to the movies directed by that director and df_i^n is the number of movies directed by d_i^n that have been rated by the user. For example, the element $(Peter.Jackson, 8.7, 3)$ specifies that the user has rated 3 movies where *Peter Jackson* has participated, with an average rating of 8.7. The remaining two preferences sets A^i and G^i follow the same representation. Therefore, A^i is formalized as the set

$$A^i = \{(a_i^1, ar_i^1, af_i^1), \dots, (a_i^n, ar_i^n, af_i^n), \dots\}, \quad (4.7)$$

where, for an entry (a_i^n, ar_i^n, af_i^n) , the first element a_i^n identifies the actor, ar_i^n is the average rating given by the user to the movies where the actor participated and af_i^n is the number of movies acted by a_i^n that have been rated by the user. Finally, G^i is similarly formalized as the set

$$G^i = \{(g_i^1, gr_i^1, gf_i^1), \dots, (g_i^n, gr_i^n, gf_i^n), \dots\}, \quad (4.8)$$

where, for an entry (g_i^n, gr_i^n, gf_i^n) , the first element g_i^n identifies the genre, gr_i^n is the average rating given by the user to the movies of that genre and gf_i^n is the number of movies of genre g_i^n that have been rated by the user. Note that dr_i^n , ar_i^n and gr_i^n are all in the interval $[1, 10]$, since these are estimated from ratings between 1 and 10.

4.2.2 Discovering User Profile Variables

In the user profile represented by Equation 4.5, the values T^i and $bias^i$ represent the user-specific quality standards. More specifically, the user threshold T^i represents the minimum rating that the user considers *good*, while the user bias $bias_i$ represents the harshness or softness of the user when rating movies. Similarly to the user preferences, these values are not specified by the user and are, therefore, estimated by leveraging on the previous ratings that the user has given. The user threshold T^i is obtained by calculating the average of the user ratings in ur_i , formalized by the equation

$$T^i = \frac{\sum_{r_i^k \in ur_i} r_i^k}{K}, \quad (4.9)$$

where K is the total number of ratings given by the user. The resulting user threshold T^i is a value such that $T^i \in [1, 10]$ and is used to filter recommended movies, i.e. remember Equation 4.4, where a movie is recommended to a user if the corresponding predicted user-movie rating is higher or equal than T^i . However, notice that a very small number of rated movies is not enough to accurately estimate a user threshold. For example, if a user has only rated one movie with a rating of 10, it would be assumed that the user only considers *good* the movies that are rated with 10. Considering this, when the number of rated movies $K < 10$, the user threshold T^i is set to 5, as it is the middle point of the rating scale.

In turn, the user bias $bias_i$ accounts for the deviation of the user ratings from the general average rating of the rated movies,

$$bias^i = \frac{\sum_{r_i^k \in ur_i} (r_i^k - avg_{<k>})}{K}, \quad (4.10)$$

where $avg_{<k>}$ is the average IMDb rating of the movie $m_{<k>}$. Note that $bias_i$ can be positive or negative, depending on the softness or harshness of the user, such that $bias^i \in [-10, 10]$. When $K < 10$, the user bias cannot be accurately estimated, so $bias^i = 0$.

4.3 Formal Model

The formal model of the implemented method starts by exploring the similarity between the user profile (Equation 4.5) and the movie profile (Equation 4.1) in order to estimate a user-movie rating based solely on movie characteristics and user preferences. A first step starts by predicting how much the user likes each aspect of the movie separately, i.e. how much the user likes the team of directors of the movie, \hat{d}_{ij} , how much the user likes the

team of actors of the movie, \hat{a}_{ij} , and how much the user likes the combination of genres of the movie, \hat{g}_{ij} . A predicted user-movie rating $\hat{p}_{r_{ij}}$ for the user u_i and the new movie m_j can be obtained by combining \hat{d}_{ij} , \hat{a}_{ij} and \hat{g}_{ij} . Figure 4.1 illustrates this process, which is discussed next.

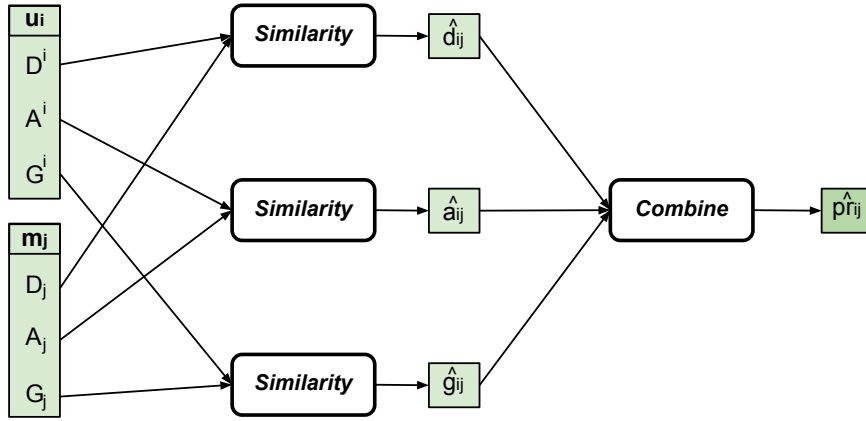


Figure 4.1: Overview of Formal Model computation.

To estimate how much the user u_i likes the team of directors of a movie m_j (\hat{d}_{ij}), the method computes the weighted average of how much the user likes each director of the movie, i.e. the weighted average of the values dr_i for each director on D_j . The weight, representing the contribution of each director rating to \hat{d}_{ij} , is pondered according to the number of movies that the user rated where the director participated, i.e. each director corresponding value df_i on the user profile D^i . The reasoning is that a user formulates a more refined and accurate opinion about a director if he/she watches more movies from that director. Hence, the directors that have been watched more times by the user are considered to have a stronger weight on the prediction. Let $D_{ij} = D^i \cap D_j$ be the set of the directors of movie m_j that are on the user profile D^i . The weight $w_{d_{ij}^n}$ of the n th director $d_{ij} \in D_{ij}$ is then obtained by the expression

$$w_{d_{ij}^n} = \frac{df_i}{\sum_{p \in D_{ij}} df_p}, \quad (4.11)$$

such that $\sum_n w_{d_{ij}^n} = 1$. Considering this, the preference of user u_i towards the team of directors of the movie m_j is obtained by the expression

$$\hat{d}_{ij} = \frac{\sum_{n \in D_{ij}} dr_i^n \cdot w_{d_{ij}^n}}{|D_{ij}|}, \quad (4.12)$$

where $|D_{ij}|$ is the number of directors on D_{ij} . Since all director ratings dr_{ij} are values between 1 and 10, the resulting average $\hat{d}_{ij} \in [1, 10]$. Note that when none of the directors of movie m_j are on the user directors set D^i , $\hat{d}_{ij} = 0$, i.e. the user does not know any of the movie directors.

How much the user u_i likes the actors of the movie m_j is obtained similarly to how \hat{d}_{ij} is obtained, i.e. by applying the same procedure. Let $A_{ij} = A^i \cap A_j$ be the set of actors of movie m_j that are on the user profile A^i . Thus, the user u_i preference towards the actors of the movie m_j is obtained by the expression

$$\hat{a}_{ij} = \frac{\sum_{n \in A_{ij}} ar_i^n \cdot w_{a_{ij}}^n}{|A_{ij}|}, \quad (4.13)$$

where $|A_{ij}|$ is the number of actors on A_{ij} and $w_{a_{ij}}^n$ is the weight of the actor a_{ij}^n . Similarly to \hat{d}_{ij} , when none of the actors of movie m_j are on the user actors A^i , $\hat{a}_{ij} = 0$. In turn, let $G_{ij} = G^i \cap G_j$ be the set of the genres of movie m_j that are on the user profile G^i . How much the user u_i likes the genres of the movie m_j is obtained by the expression

$$\hat{g}_{ij} = \frac{\sum_{n \in G_{ij}} gr_i^n \cdot w_{g_{ij}}^n}{|G_{ij}|}, \quad (4.14)$$

where $|G_{ij}|$ is the number of genres on G_{ij} and $w_{g_{ij}}^n$ is the weight of the genre g_{ij}^n . Like \hat{d}_{ij} and \hat{a}_{ij} , $0 \leq \hat{g}_{ij} \leq 10$, with 0 occurring when none of the movie genres are on the user genres G^i .

After estimating how much the user likes each aspect of the movie separately (i.e. \hat{d}_{ij} , \hat{a}_{ij} and \hat{g}_{ij}), a rating prediction is obtained by combining these values. Let T be the number of feature set ratings \hat{d}_{ij} , \hat{a}_{ij} and \hat{g}_{ij} that are different from 0. The predicted rating $\hat{p}r_{ij}$ for user u_i and the *cold-start* movie m_j is then obtained by the equation

$$\hat{p}r_{ij} = \frac{1}{T} (\theta_d \cdot \hat{d}_{ij} + \theta_a \cdot \hat{a}_{ij} + \theta_g \cdot \hat{g}_{ij}), \quad (4.15)$$

where θ_d , θ_a and θ_g are constants controlling the contributions of directors, actors and genres to the rating prediction: it is argued that the different aspects of the movie influence how much the user will like that movie differently. For example, a user might dislike *action* movies in general but still like a certain action movie where *Angelina Jolie* participates if the user likes that actress. In this example, actors seem to present a higher relevance than genres, so $\theta_a > \theta_g$. The optimal values for these constants will be estimated in Chapter 5.

Here it is argued that there are two main characteristics that a recommended new movie should have: first, it should match the user preferences; second, it should not be a low quality movie. Note that $\hat{p}r_{ij}$ lacks the second component, as it does not take into account the inherent quality of the movie or of its components. In the following sections, the computation of $\hat{p}r_{ij}$, i.e. Equation 4.15, is extended to include Social-Media information and improve rating predictions.

4.4 Social-Media Trends and Reputations

In this section, the Social-Media feedback of a movie m_j , formalized in Equation 4.1, is formalized as the set

$$S_j = \{T(m_j), \text{reps}(m_j)\}, \quad (4.16)$$

containing a set of tweets $T(m_j)$ where the movie m_j is mentioned and the reputation $\text{reps}(m_j)$ of all directors and actors participating in m_j . The contents of this section link to the monitoring processes discussed in Chapter 3.

4.4.1 Popularity of New Movies on Twitter

The Social-Media feedback about a new movie is obtained from Twitter, as described in Section 3.3: tweets where the movie title is identified are stored and labelled according to the movie name. The captured tweets are then classified by a k -NN sentiment classifier such that, for each tweet, it is inferred if it is a positive, negative or neutral reference to the movie. A tweets index is then constructed to allow fast look-ups by movie title, in order to retrieve the tweets referring it. Formally, the retrieved tweets for a certain movie m_j are represented as the set

$$T(m_j) = \{(t_{j1}, s_{j1}), \dots, (t_{jl}, s_{jl}), \dots, (t_{jM}, s_{jM})\}, \quad (4.17)$$

where t_{jl} is the tweet (talking about m_j) and s_{jl} is the sentiment of the tweet such that $s_{jl} \in \{\text{pos}, \text{neg}\}$. The tweets referring m_j neutrally are discarded, as those are considered to not express any sentiment towards the movie.

Krauss et al. [KNSFG08] has showed that movie trendiness is projected in *The Oscar* nominations, which are generally associated with highly rated movies. The set $T(m_j)$, containing tweets targeting movie m_j , can be used to predict its trendiness. Oghina et al. [OBTR12] have shown that the fraction of likes/dislikes is the strongest feature for predicting IMDb movie ratings from Social-Media. Following this remarks, the popularity of a movie m_j is measured by the equation

$$\text{pop}(m_j) = \frac{|pos_{m_j}|}{|T(m_j)|}, \quad (4.18)$$

where $|pos_{m_j}|$ is the number of positive tweets referring the movie m_j and $|T(m_j)|$ is the total number of tweets referring m_j .

4.4.2 Reputation of Directors and Actors on IMDb

The Social-Media feedback on directors and actors is obtained from IMDb, as described in Section 3.2: movie reviews are crawled and used to build a sentiment graph linking named-entities, from which the reputation of directors and actors is computed [PSM14a].

This step allows the reputation of the directors and actors of the new movies to be obtained. Formally, the reputation of all the directors and actors participating on movie m_j is retrieved as the set

$$\text{reps}(m_j) = \{\text{rep}(e_1), \dots, \text{rep}(e_k), \dots\}, \quad (4.19)$$

where the reputation of each entity e_k is $\text{rep}(e_k) \in [0.0, 1.0]$, with 0.0 being the worst reputation and 1.0 being the best reputation. Formally, $\forall e_k \in D_j \cup A_j, \text{rep}(e_k) \in \text{reps}(m_j)$.

4.5 Recommendation with Social-Media Signals

Moshfeghi et al. [MPJ11] and Krauss et al. [KNSFG08] obtained hidden latent factors to correlate movies through sentiment analysis. Here, however, new movies do not have reviews and tweets about new movies are too scarce to infer relevant latent topics. Therefore, the emotion expressed towards movies is explored as a qualitative measure, in which the inherent quality of new movies, directors and actors is obtained and considered.

The improved rating prediction \hat{r}_{ij} is therefore obtained by considering both how popular the movie is, $\text{pop}(m_j)$, and how much a user might enjoy the movie m_j , given the reputation $\text{reps}(m_j)$ of its participants. The described approach is formalized as

$$\hat{r}_{ij} = \alpha_t \cdot (\text{pop}(m_j) + \text{bias}^i) + (1 - \alpha_t) \cdot \hat{p}_{ij|\text{reps}(m_j)}, \quad (4.20)$$

where α_t is a constant reflecting the importance of the movie popularity to the final user-movie rating \hat{r}_{ij} . Note that $\text{pop}(m_j)$, obtained from Twitter, is a representation of the general opinion towards the new movie m_j . However, different users have different standards when compared to the general public. Hence, the user bias bias^i (obtained by Equation 4.10) is used to model the general opinion about the movie to the user standards. How much the user likes the characteristics of the movie given the reputation of its participants, $\hat{p}_{ij|\text{reps}(m_j)}$, is an extension of \hat{p}_{ij} . By considering the reputation information in $\text{reps}(m_j)$, Equation 4.15 is extended to

$$\hat{p}_{ij|\text{reps}(m_j)} = \frac{1}{T} (\theta_d \cdot \hat{d}_{ij|\text{reps}(m_j)} + \theta_a \cdot \hat{a}_{ij|\text{reps}(m_j)} + \theta_g \cdot \hat{g}_{ij}), \quad (4.21)$$

where $\hat{d}_{ij|\text{reps}(m_j)}$ is an extension of \hat{d}_{ij} and $\hat{a}_{ij|\text{reps}(m_j)}$ is an extension of \hat{a}_{ij} . Figure 4.2 illustrates the described process as an extension of Figure 4.1. The calculus of \hat{d}_{ij} , \hat{a}_{ij} and \hat{g}_{ij} is omitted in order to simplify the illustration.

4.5.1 Modeling User Preferences with the Reputation of Entities

Up until this point, when predicting the values \hat{d}_{ij} and \hat{a}_{ij} (i.e., how much a user likes or dislikes the directors and actors of a movie), the entities that the user does not know were not considered. However, here it is argued that these entities also influence how

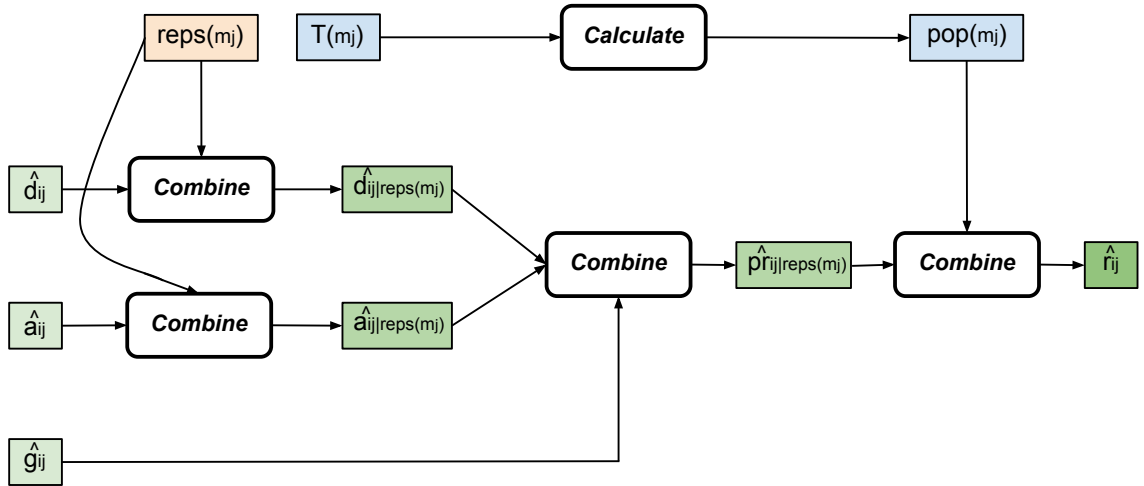


Figure 4.2: Overview of the complete rating prediction model.

the user will ultimately rate the movie. Hence, the calculation of \hat{d}_{ij} and \hat{a}_{ij} is enhanced, given the reputations of directors and actors available in $reps(m_j)$. For this purpose, two new variables, $\hat{u}d_{ij}$ and $\hat{u}a_{ij}$, are introduced to express the reputation of the unknown directors and actors,

$$\hat{u}d_{ij} = \frac{\sum_{d \in D_j - D^i} rep(d)}{|D_j - D^i|}, \quad \hat{u}a_{ij} = \frac{\sum_{a \in A_j - A^i} rep(a)}{|A_j - A^i|}, \quad (4.22)$$

where $D_j - D^i$ and $A_j - A^i$ are the sets of directors and actors on movie m_j that the user does not know.

To consider $\hat{u}d_{ij}$ and $\hat{u}a_{ij}$ in the calculation of $\hat{p}r_{ij|reps(m_j)}$, one ought to note that \hat{d}_{ij} and \hat{a}_{ij} represent user preferences towards their known directors and actors. Thus, $\hat{d}_{ij|reps(m_j)} = \hat{d}_{ij}$ and $\hat{a}_{ij|reps(m_j)} = \hat{a}_{ij}$ when all the directors or actors of m_j are known by the user, and $\hat{d}_{ij|reps(m_j)} = \hat{u}d_{ij}$ and $\hat{a}_{ij|reps(m_j)} = \hat{u}a_{ij}$, when the user does not know any directors or actors of the movie. The general case is when the user knows some of the directors and actors of the movie. Formally, the final directors and actors scores $\hat{d}_{ij|reps(m_j)}$ and $\hat{a}_{ij|reps(m_j)}$ are calculated by considering both the user preferences and the public opinion, i.e. a weighted average between the scores of the known entities and the unknown entities,

$$\hat{d}_{ij|reps(m_j)} = \delta_{ud} \cdot (\hat{u}d_{ij} + bias^i) + (1 - \delta_{ud}) \cdot \hat{d}_{ij}, \quad (4.23)$$

$$\hat{a}_{ij|reps(m_j)} = \delta_{ua} \cdot (\hat{u}a_{ij} + bias^i) + (1 - \delta_{ua}) \cdot \hat{a}_{ij}, \quad (4.24)$$

where the constants δ_{ud} and δ_{ua} represent the contribution of the unknown directors and actors to the computation of $\hat{d}_{ij|reps(m_j)}$ and $\hat{a}_{ij|reps(m_j)}$ respectively. They are computed

as

$$\delta_{ud} = \frac{|D_j - D^i|}{|D_j|}, \quad \delta_{ua} = \frac{|A_j - A^i|}{|A_j|}, \quad (4.25)$$

where $|D_j - D^i|$ is the number of directors on movie m_j that the user u_i does not know and $|A_j - A^i|$ is the number of actors on movie m_j that the user does not know. The user bias $bias^i$ is, once again, used to model the general opinion on directors and actors to the user standards.

4.6 Summary

This chapter presented the recommendation algorithm for *cold-start* movies, which leverages on the Social-Media information collected in Chapter 3. A recommendation problem starts with a user u_i , described only by his previous ratings ur_i , and a set of movies M , containing rated movies and *cold-start* movies. The rated movies are used together with the user ratings to discover user preferences, while the *cold-start* movies are potentially recommended. The most important aspects of the implemented method are:

- The reputation of a movie directors and actors are used to extend \hat{d}_{ij} and \hat{a}_{ij} , which rank how much the user likes the team of directors and actors of a new movie, into considering the directors and actors that the user does not know, resulting in the values $\hat{d}_{ij|reps(m_j)}$ and $\hat{a}_{ij|reps(m_j)}$. By combining these values with \hat{g}_{ij} , a rating prediction based on the user-movie similarity $\hat{p}r_{ij|reps(m_j)}$ is obtained;
- The popularity of a movie on Twitter, $pop(m_j)$, is combined with the user-movie similarity $\hat{p}r_{ij|reps(m_j)}$ to generate a final rating prediction \hat{r}_{ij} . As a result, the final rating prediction \hat{r}_{ij} considers both how the movie matches the user preferences and how good the movie is as a whole;
- A movie is finally recommended if the predicted user-movie rating \hat{r}_{ij} is above the user-specific quality threshold.



Results and Evaluation

This chapter discloses the experiments and results regarding the evaluation of the method implemented in Chapter 4. First, the used dataset is described together with its extraction process. The evaluation methodology is then presented, where both the baselines and evaluation measures are specified. By the end of the chapter, the various aspects of the method are tested against the baselines and evaluated.

5.1 Dataset

To perform a thorough evaluation of the developed method, a dataset comprising of user-movie ratings and tweets regarding movies is needed. Moreover, movies metadata and IMDb reviews are also necessary in order to compute full-fledged rating predictions. While publicly available datasets containing the required data already exist, such as the MovieLens¹ and Netflix² datasets for user-movie ratings and the MovieTweatings [DDPM13] dataset for tweets regarding movies, they are not compatible with each other and with the considered scenario. For example, while MovieLens and MovieTweatings contain extensive numbers of user-movie ratings and tweets respectively, they were not collected in the same time interval: as a result, they do not concern the same movies. Furthermore, MovieTweatings comprises tweets of users sharing their IMDb user-movie ratings, contradicting the scenario at hand, where tweets are collected for movies that have not been rated on IMDb.

To set up a realistic evaluation scenario, a dataset was collected by focusing on a total of 60 new movies, finalists on 5 popular movie awards ceremonies: the 2014 editions

¹www.movielens.org

²www.netflixprize.com

of *The Golden Globes*, *The Critic's Choice Awards*, *The BAFTA Film Awards*, *The Independent Spirit Awards* and *The Oscars*. These events occurred between January and March, designating movies that have been released in 2013. The movies were selected so a great number of relevant tweets could be captured in small time period. Figure 5.1 relates the award ceremonies and the tweets extraction process to a simplistic timeline, covering the elected time interval. A total of 52,236 tweets referring the selected movies were captured, averaging 870 tweets per movie.

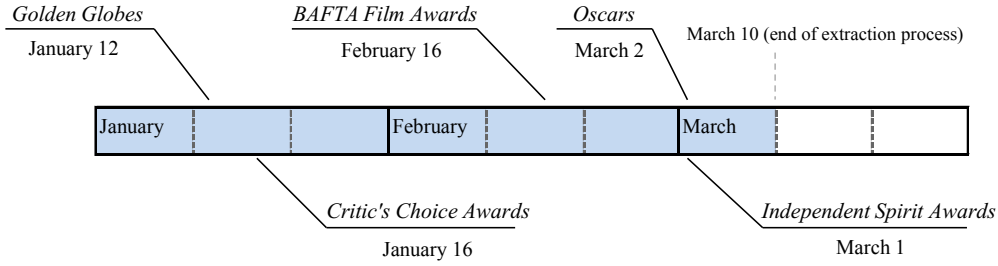


Figure 5.1: Timeline for Twitter extraction process.

The user-movie ratings extraction process focused on users who had rated at least one of the 60 new movies on IMDb. This guideline is essential, since user-movie ratings given to the new movies are used as the testset in Section 5.3. Therefore, the 500 last users who have rated each of the new movies were selected as the target users. For each target user, all the given ratings are collected: ratings given to old movies are necessary to discover user preferences. While collecting users and ratings, users who have not rated any movie besides the 60 new movies are discarded. Algorithm 2 summarizes the described extraction process.

Algorithm 2 IMDb Ratings Extractor

Input: $New_movies \leftarrow$ The list of selected new movies.

Output: $User_ratings \leftarrow$ The list of users and respective movie ratings.

```

1: for all  $m_i \in New\_movies$  do
2:    $Movie\_users_i =$  Get the list of 500 users who last rated  $m_i$ .
3:   for all  $u_j \in Movie\_users_i$  do
4:     if  $u_j$  not in  $User\_ratings$  then
5:        $Ratings_j =$  Get all movies rated by  $u_j$  and respective ratings.
6:       if  $Ratings_j$  contains movies that are not in  $Movies$  then
7:          $User\_ratings_j = Ratings_j$ 
8:       end if
9:     end if
10:  end for
11: end for
12: return  $User\_ratings$ 

```

A total of 1,064,766 ratings were collected, given by 2,909 users to the 60 new movies and 46,843 old movies. The new ratings amount to 27,394, corresponding to approximately 2.6% of all ratings. Figure 5.2 plots information regarding the distribution of the

obtained user-movie ratings: 5.2a shows the frequency of each rating score on the dataset and 5.2b plots the distribution of users per threshold value. Approximately a quarter of the collected ratings are positive, i.e. greater or equal to 6, with 7 being the most common rating from the old ratings (22.7%) and 8 being the most frequent from the new ratings (26.8%). Overall, the distribution pattern of both old ratings and new ratings is very similar. Furthermore, the great frequency of high old ratings translated into generally high user thresholds: while they vary between 1 and 10, the average user threshold is 7.24, signifying that only high rating predictions are translated into recommendations.

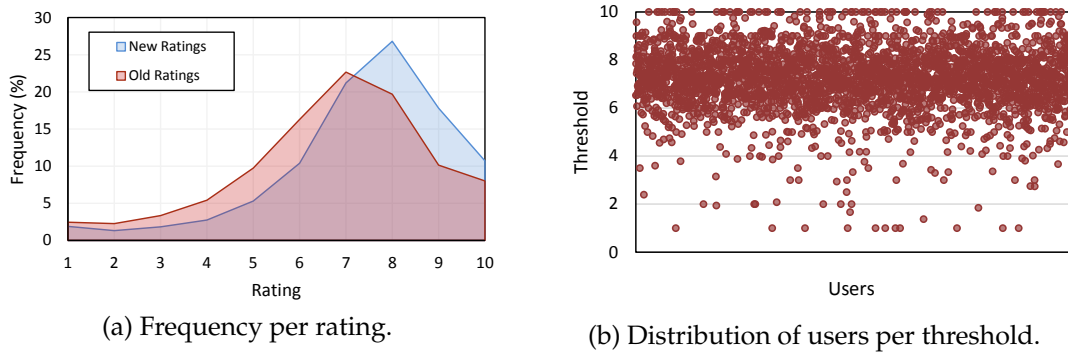


Figure 5.2: Dataset information on user-movie ratings.

The metadata for all the 46,903 rated movies was obtained through the publicly available OMDb API³ which, in turn, crawls IMDb for data on movies. While the OMDb API collects a variety of information about the target movies, only the relevant information was stored, namely the list of directors, actors and genres. Lastly, IMDb reviews were crawled in order to compute the reputation [PSM14a] of the new movies' directors and actors. The extraction process focused on the reviews for the movies where the targeted directors and actors have participated, i.e. these reviews have a high chance of referring them. However, only reviews for movies released between 2008 and 2012 are stored, as it is argued that the reputation of entities can change drastically over time. A total of 124,236 reviews were collected, corresponding to the 225 actors and 169 directors who have participated in the 60 new movies. Figure 5.3 summarizes the resulting dataset while relating the obtained data to the scenario entities.

5.2 Methodology

The experiments were performed by leveraging on the collected dataset, described in 5.1. More specifically, the dataset was split in two sets: the test set, comprising the user-movie ratings given to the 60 new movies; the training set, comprising all the other information, namely the old ratings, the metadata, the tweets and the reviews. The goal was to use the training set, containing the users past ratings and the Social-Media information, to

³www.omdbapi.com

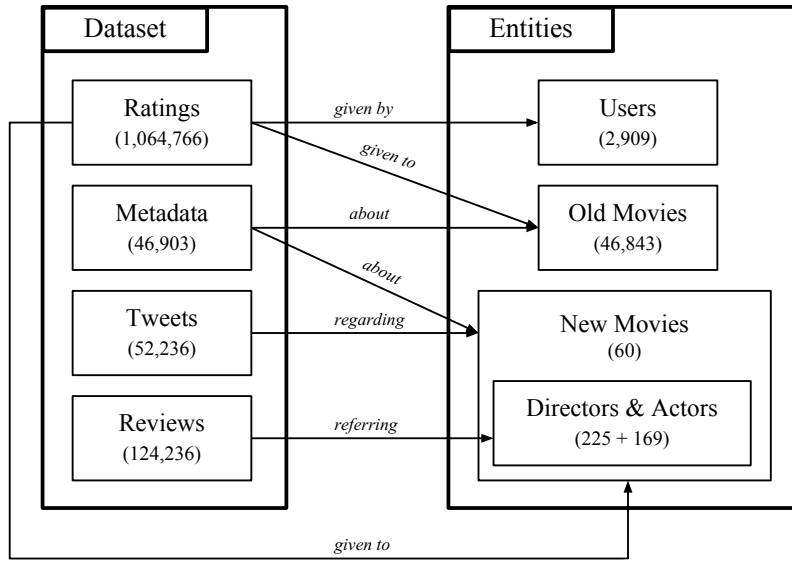


Figure 5.3: Summary of dataset.

compute user-movie rating predictions for the 60 new movies. In turn, the test set was used to evaluate the applied method, by comparing the predicted and the real ratings given to those movies. The described process is illustrated in Figure 5.4.

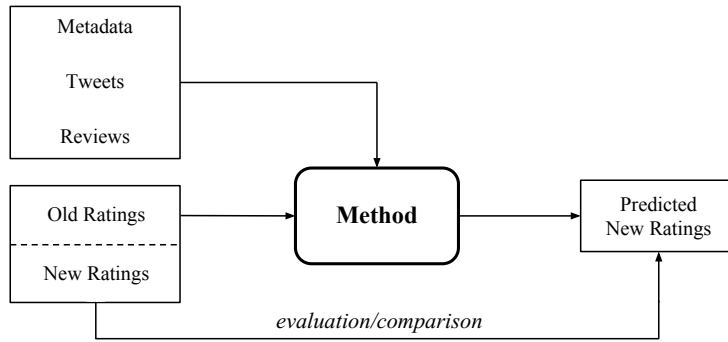


Figure 5.4: The evaluation methodology.

To assess the different aspects of the implemented method, rating predictions were computed and evaluated for three different variants of Equation 4.20:

- **MRep (Movie Reputation):** This first variant assessed the contribution of the movie popularity $pop(m_j)$, which is inferred from tweets regarding the new movie m_j , to improve rating predictions. In this variant, the predicted rating \hat{r}_{ij} for an user u_i and a new movie m_j is formally obtained by the equation

$$\hat{r}_{ij} = \alpha_t \cdot (pop(m_j) + bias_i) + (1 - \alpha_t) \cdot \hat{p}r_{ij}, \quad (5.1)$$

which is equivalent to Equation 4.20, except for considering the $\hat{p}r_{ij}$ variable instead of $\hat{p}r_{ij|reps(m_j)}$: the reputation of directors and actors is not accounted.

- **ERep (Entities Reputation):** This second variant assesses the contribution of the reputation of directors and actors, $\hat{d}_{ij|reps(m_j)}$ and $\hat{a}_{ij|reps(m_j)}$ respectively, which are calculated from written reviews. In ERep, the final predicted rating is $\hat{p}r_{ij|reps(m_j)}$, as defined by Equation 4.21. Antithetically to MRep, this variant does not consider the popularity of the movie.
- **FRep (Full Reputation):** The third method uses the full spectrum of Social-Media information, where both the popularity of the movie and the reputation of its entities are considered, i.e., the rating prediction is obtained by the original Equation 4.20.

The different variants of the method are compared to three baseline methods, where ratings are predicted by leveraging only on movies metadata and past ratings:

- **k-NN (k-Nearest Neighbour):** The first baseline is the k-NN algorithm which is widely successful for hybrid recommendations [MMN02; ALPKO09]. In k-NN, a movie-feature matrix was built for each user separately, containing his rated movies: each movie is a binary vector representing the participating directors, actors and the respective genres. Additionally, each movie is labelled by its user-movie rating. The Manhattan distance was used to find the k most similar rated movies to the new movies, with k being equal to $\frac{1}{4}$ of the number of movies the user has rated. A predicted rating for a new movie is obtained by the most occurring label on the k most similar rated movies.
- **FM1 (Formal Model 1):** The second baseline is the formal model of the implemented method, where ratings are predicted without Social-Media feedback, i.e. with Equation 4.15. In FM1, θ_d , θ_a and θ_g are all equal to 1: the directors score, actors score and genres score all contribute equally to the predicted rating $\hat{p}r_{ij}$.
- **FM2 (Formal Model 2):** Similarly to FM1, the final baseline also obtains rating predictions with Equation 4.15, but with $\theta_d = 0.35$, $\theta_a = 0.20$ and $\theta_g = 0.45$. These values were estimated to hold the best results in 5.3.2.

Figure 5.5 summarizes the different evaluated methods and baselines, by illustrating the input and output data of each method.

To quantify the obtained results, two widely popular metrics are used: the Mean Average Error (MAE) and the F-Measure (F-M). The MAE is calculated from the absolute error between the predicted user-movie ratings and the real ratings. Let R be the set containing the n real user-movie ratings for the new movies. The MAE of a method is

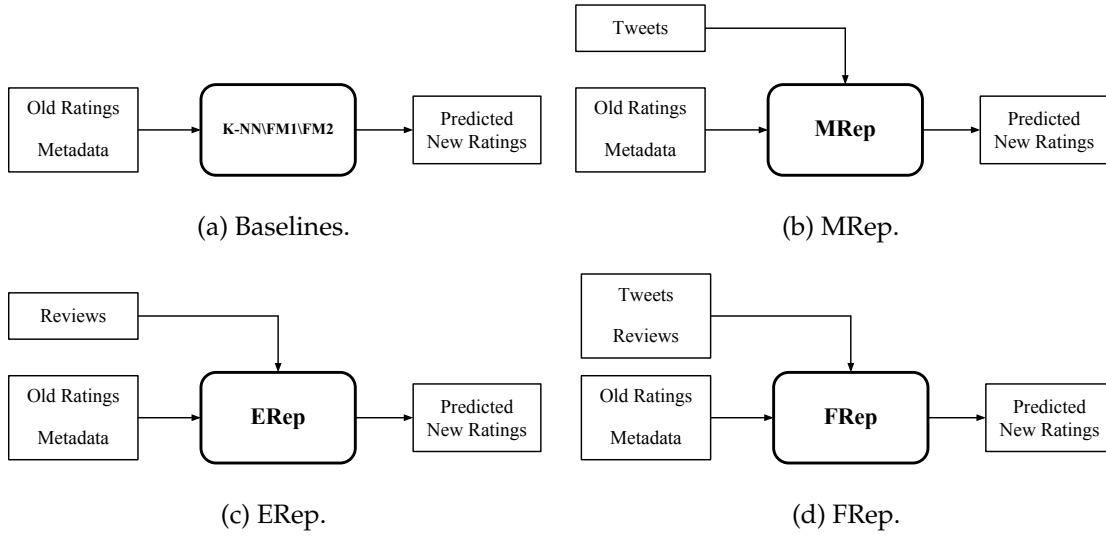


Figure 5.5: The evaluated methods.

formally obtained by the expression

$$MAE = \frac{1}{n} \cdot \sum_{r_{ij} \in R} abs(\hat{r}_{ij} - r_{ij}), \quad (5.2)$$

where r_{ij} is the real rating from user u_i to the movie m_j and \hat{r}_{ij} is the respective predicted rating. The F-Measure, in turn, is calculated to assess how well the rating predictions translate into recommendations. Let rec be the total number of recommended new movies, where the predicted rating \hat{r}_{ij} is greater than the respective user threshold T_i , i.e. $\hat{r}_{ij} > T_i$. Furthermore, let rel be the total number of relevant new movies, where the real rating r_{ij} is greater than the respective user threshold T_i , i.e. $r_{ij} > T_i$. The Precision and Recall measures are formally obtained by the equations

$$precision = \frac{rec \cap rel}{rec} \quad recall = \frac{rec \cap rel}{rel}. \quad (5.3)$$

The Precision measure qualifies the ability of the method in distinguishing relevant and irrelevant movies. Differently, the Recall measure qualifies the ability of the method in identifying relevant movies, i.e. false positives are not considered. The F-Measure is obtained by calculating the harmonic mean of the Precision and Recall measures, formally defined by the equation

$$F_M = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (5.4)$$

Before evaluating the main method, Twitter is assessed as a source of reliable movie feedback by calculating the MAE between the predicted popularity $pop(m_j)$ and the respective average IMDb rating for all the new movies. Secondly, the best values for θ_d , θ_a , θ_g and α_t are estimated, so the methods can be tested to their full potential. Finally,

all the variants of the main approach are tested against the baselines, by comparing the obtained MAE and F-M results.

5.3 Results and Discussion

5.3.1 Twitter for Estimating Movie Popularity

In Equation 4.18, Twitter is used as a source of movie feedback to predict the popularity for new movies. The predicted popularity ratings are compared with the average IMDb ratings of the target movies, captured several months after the movies' release dates (at the start of July 2014). Figure 5.6 plots the predicted ratings and the IMDb average ratings. From it, the overall deviation of the predicted ratings can be observed in order to draw relevant conclusions.

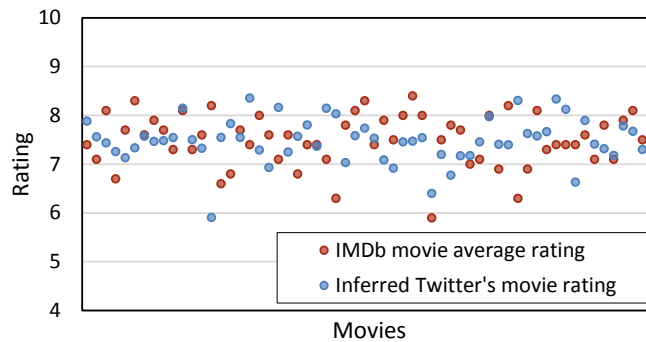


Figure 5.6: Twitter-based Movie Ratings vs IMDb Movie Ratings.

The overall obtained MAE is 0.59, which is in the same error range found in literature [OBTR12], where the best obtained MAE using Twitter is 0.51. The prediction errors varied from 0.026 (*Blue is the Warmest Colour*) to 2.29 (*Her*). By analysing the overall error deviations, it can be observed that movies with lower IMDb ratings are more likely to have a higher prediction error: for instance, while *Blue is the Warmest Colour* has an average IMDb rating of 8.0, examples of high error such as *The Invisible Woman* (MAE = 2.01) and *Computer Chess* (MAE = 1.73) have an average IMDb rating of 6.3. These results suggest that Twitter users are more likely to share positive tweets about movies than negative tweets, making Equation 4.18 more precise for highly rated movies. The movie *Her* holds the only observable exception to this conclusion: while it is a highly rated movie (IMDb Rating = 8.2), it has obtained the largest prediction error. A possible reason for this might be its ordinary movie title, which makes it more likely to capture unrelated tweets.

5.3.2 Estimation of the θ_d , θ_a and θ_g Parameters

The parameters θ_d , θ_a and θ_g control the contribution of directors, actors and genres for rating predictions, respectively (Equations 4.15 and 4.21). These values express the idea that each of these movie components have a different influence on the user-movie ratings and should be considered when predicting new ratings. In order to estimate the best values for θ_d , θ_a and θ_g , rating predictions are computed with Equation 4.15 for different combinations of values. Figure 5.7 plots the MAE and F-Measure curves by focusing on each parameter separately, where the value of the remaining parameters is the same. For example, for the $\theta_d = 0.70$ point, $\theta_a = \theta_g = 0.15$. These curves let us have an overall idea of the influence of each parameter on the results.

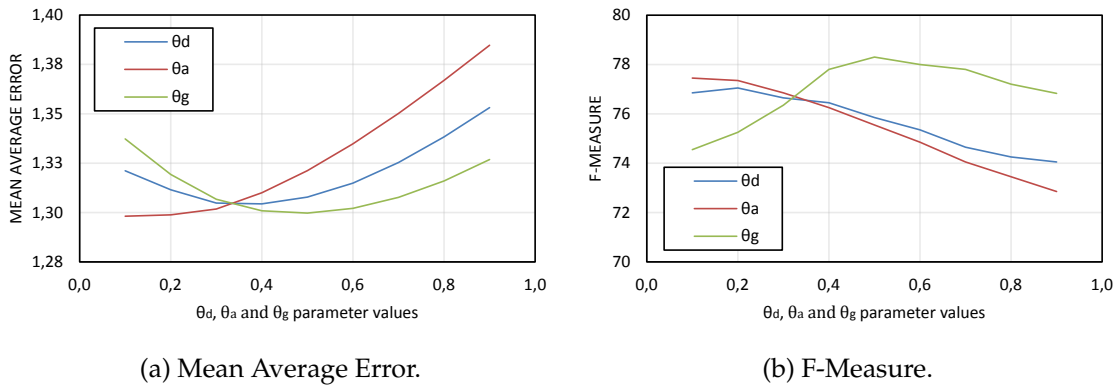


Figure 5.7: Estimation of θ_d , θ_a and θ_g as a function of MAE and F-Measure.

The parameter θ_g presents the best results: when $\theta_g < 0.30$ the method obtains both the worst MAE and F-Measure results, while the best results are obtained when θ_g is higher than both θ_d and θ_a . These results show that the genre of the movie has more influence on the predicted ratings, when compared to the directors and actors. However, when θ_g is considered too much, the results start to deteriorate, meaning that the other parameters are also relevant. Interestingly, θ_a seems to have the least influence on the user-movie ratings: while the actors are probably the main reason why people watch certain movies, the directors and the genres seem to have a higher influence in the overall quality of the movie. To estimate the exact best values for each parameter, 10-fold cross validation was used to predict rating for various the combinations of values where $0.05 \leq \theta_d, \theta_a, \theta_g \leq 0.95$. The best estimated values were $\theta_d = 0.35$, $\theta_a = 0.20$ and $\theta_g = 0.45$, where the MAE was 1.2962 and the F-Measure was 79.6%.

5.3.3 Estimation of the α_t Parameter

The parameter α_t controls the influence of the movie popularity rating in its user-movie rating predictions, expressed by Equation 4.20. In order to find the best value for α_t , rating predictions are computed for various α_t values on a validation dataset, containing 300 randomly selected users from the main dataset (Section 5.1). Figure 5.8 plots the MAE

and F-Measure curves for a range of α_t value on MRep and FRep.

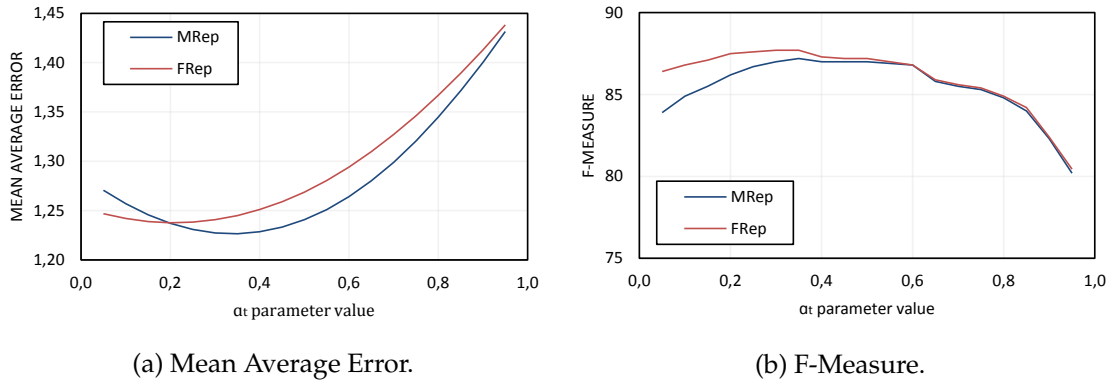


Figure 5.8: Estimation of α_t as a function of MAE and F-Measure.

Both MRep and FRep present the best results for α_t values below or equal to 0.40 – after this point, both the MAE and F-Measure start to deteriorate. For MRep, both the best MAE and F-Measure values are obtained at $\alpha_t = 0.35$ (MAE = 1.2266 and F-Measure = 87.2%). For FRep, the best F-Measure is also obtained at $\alpha_t = 0.35$ (87.7%), while the best MAE is obtained at $\alpha_t = 0.20$. These results suggest that the popularity of a movie has significant importance when predicting user-movie ratings for *cold-start* movies. However, if the popularity of the movie is considered too much against users personal preferences (i.e., α_t is too high), the predicted user-movie rating loses the personalization component, leading to less accurate predictions. For subsequent experiments, $\alpha = 0.35$ as it is estimated to be the best performing value.

5.3.4 Influence of User Bias

User bias is considered in Equations 4.20, 4.23 and 4.24 to adjust the general opinions about movies, directors and actors to each user standards. Figure 5.9 shows the density of users per bias value. From it, the overall divergence of users bias from the general opinion can be estimated and discussed.

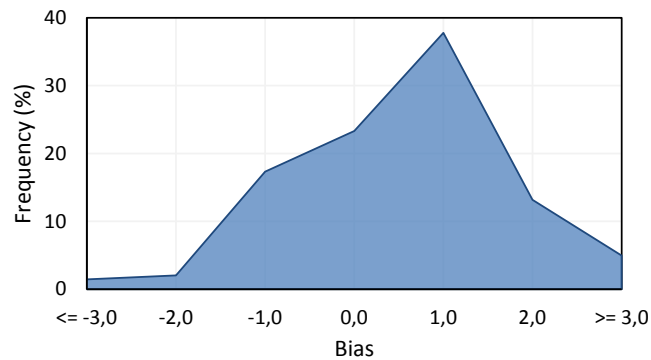


Figure 5.9: Distribution of user bias.

The bias values of the dataset users vary from -7.0 to 4.7, with a mean absolute bias of 0.98. This means that, in average, the opinion of a user is estimated to differ 0.98 rating values from the general opinion. Furthermore, 39.2% of the users have an absolute bias of 1.0 rating or more, enforcing the need of adjusting the public opinion to each user. To assess the influence of user bias, user-movie ratings are computed with the three main methods (MRep, ERep and FRep), both considering and not considering user bias ($bias_i = 0$). Figure 5.10 presents the obtained MAE and F-Measure results for each method, both including and excluding user bias.

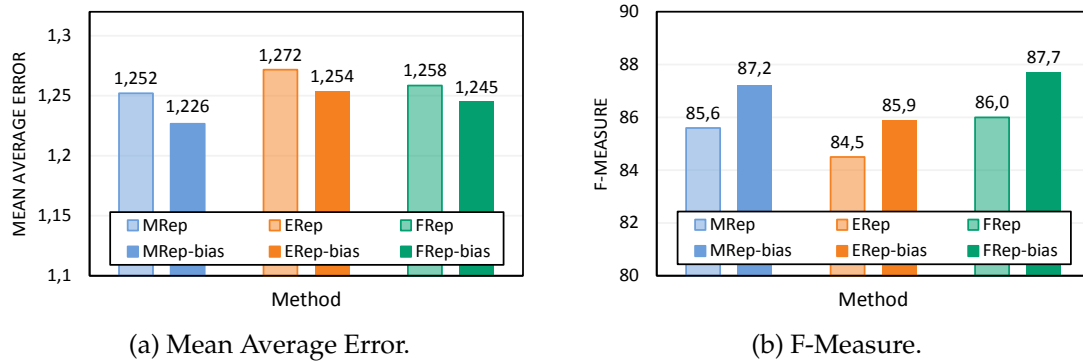


Figure 5.10: The influence of user bias.

As can be observed, the inclusion of user bias improves the results in all three approaches: the MAE decreases in all methods and the F-Measure increases. In terms of rating prediction, MRep presents the best results (MAE = 1.226) and ERep presents the worst results (MAE = 1.254), while FRep presents neither the best nor the worst result (MAE = 1.245). These values suggest that the popularity of the movie itself is more useful for predicting the user-movie rating, when compared to the reputation of its directors and actors. When computing recommendations, assessed by F-Measure, FRep presents the best results (87.7%), when compared to MRep (87.2%) and ERep (85.9%). Unlike when predicting the user-movie rating, considering both the popularity of movies and the reputation of directors and actors is the best approach for distinguishing relevant movies from irrelevant movies. For subsequent experiments, user bias is considered as it improved the results for all methods.

5.3.5 Methods Comparison

In a recommendation scenario, different users have rated different numbers of movies, e.g. in our dataset, the user with the sparsest list of watched movies has rated 9 old movies, while the user with the largest list has rated 4752. Figure 5.11 plots the MAE and F-Measure results for all considered methods, applied to users with different numbers of rated old movies. From the obtained results it is possible to compare how each method handles different levels of user sparsity.

From all the methods, K-NN presents the worst results: it has the worst MAE for all

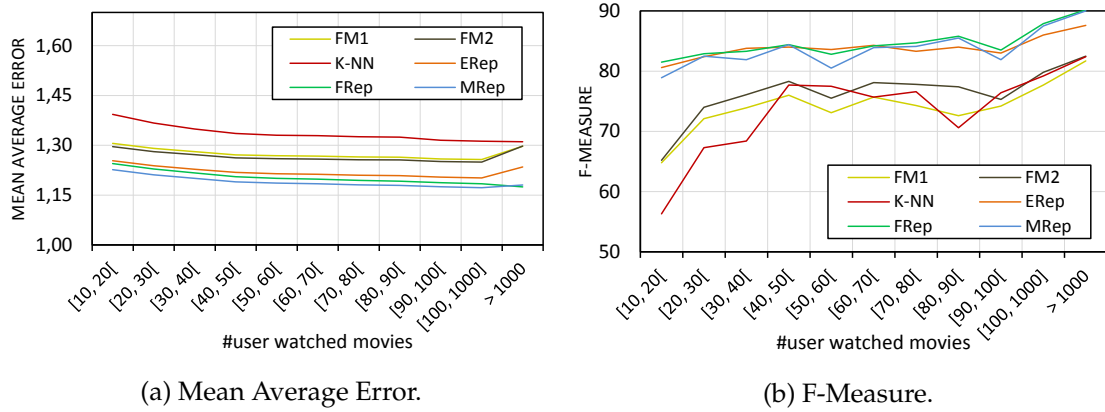


Figure 5.11: Comparative results for different numbers of watched movies.

levels of users; in recommendation, it only matches other methods for users with more than 40 rated movies. From the other baseline methods, FM2 presents better results than FM1 in both MAE and F-Measure for all values, suggesting that directors, actors and genres should weight differently when predicting user-movie ratings (i.e., θ_d , θ_a and θ_g should not be equal). The three main methods perform better than all baselines, both in rating prediction and recommendation. ERep performs better than MRep when recommending to users with less than 70 rated movies, while MRep performs better than ERep for users with a high number of rated movies. This shows that the popularity of a movie is only better for recommendation, in comparison to the reputation of directors and actors, when accompanied with well-defined user preferences. FRep presents the best recommendation results overall: it performs very closely to ERep when recommending to users with less or equal to 70 rated movies and very closely to MRep when recommending for users with a lot of rated movies. Table 5.1 summarizes the overall results on all measures for all the methods, for the complete dataset.

Method	MAE	Prec. (%)	Rec. (%)	F-M (%)
<i>k</i> -NN	1.3933	70.1	86.5	78.3
FM1	1.3058	70.3	85.2	77.8
FM2	1.2962	71.4	87.7	79.6
MRep	<u>1.2266</u>	<u>76.0</u>	98.5	87.2
ERep	1.2536	75.6	96.1	85.9
FRep	1.2450	<u>76.0</u>	<u>99.4</u>	<u>87.7</u>

Table 5.1: Overall comparative results.

Overall, all the methods that consider Social-Media information outperform the baselines. In terms of rating prediction, MRep presents the best MAE results. FRep, for instance, presents the best results in all recommendation measures, with a total F-Measure of 87.7%. This value surpasses the best baseline results, obtained with FM2, by 8.1%. The

major improvement in Recall values for the Social-Media methods (i.e., MRep, ERep and FRep) relatively to the baselines shows that the reputation of movies, directors and actors helps especially in identifying great movies, which are more usually relevant for users.

5.3.6 Cases of Extreme User Cold-Start

When users have not rated many movies, their preferences cannot be well modelled: these users suffer from the *cold-start* problem. This happens mostly, but not exclusively, for users who are new to the system. A scenario where all 2,909 users of the dataset suffer from the *cold-start* problem was simulated by not considering their old ratings. Experiments with MRep, ERep and FRep were conducted, in order to assess the performance of these methods when faced with simultaneous movie and user *cold-start*. Random recommendations are computed for comparison, as these are commonly used in literature as a baseline to evaluate methods tackling this scenario [LVLD08; PC09]. Figure 5.12 plots the results obtained in 20 independent runs with random recommendations.

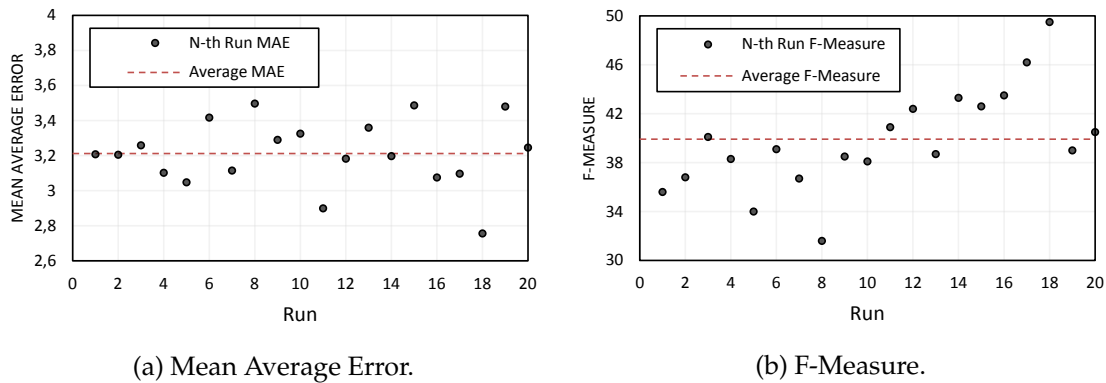


Figure 5.12: MAE and F-Measure results for random recommendations.

The random runs obtained an average MAE of 3.21 and an average F-Measure of 39.8%. At best, the approach achieved a MAE of 2.76, corresponding to an F-Measure of 49.5%. Figure 5.13 compares the results obtained for MRep, ERep and FRep with the results achieved with this baseline. Note that MRep, ERep and FRep cannot consider users bias in this scenario, since there are no previously given ratings from any user.

All the main methods outperformed the baseline: the main method with the weakest results (ERep, MAE = 1.77 and F-Measure = 63.5%) improved on the best run of the baseline approach by a MAE difference of 0.99 and an F-Measure difference of 14%. The best results were obtained by MRep, where MAE = 1.62 and F-Measure = 74.7%. These results show that the popularity of a movie is a good baseline predictor of its quality and is useful for recommending movies when the user preferences are not known. While the reputations of the movie directors and actors present much weaker results, these also prove to be an average predictor of a movie quality, as a 63.5% recommendation accuracy is very good for a scenario where there is no information on users.

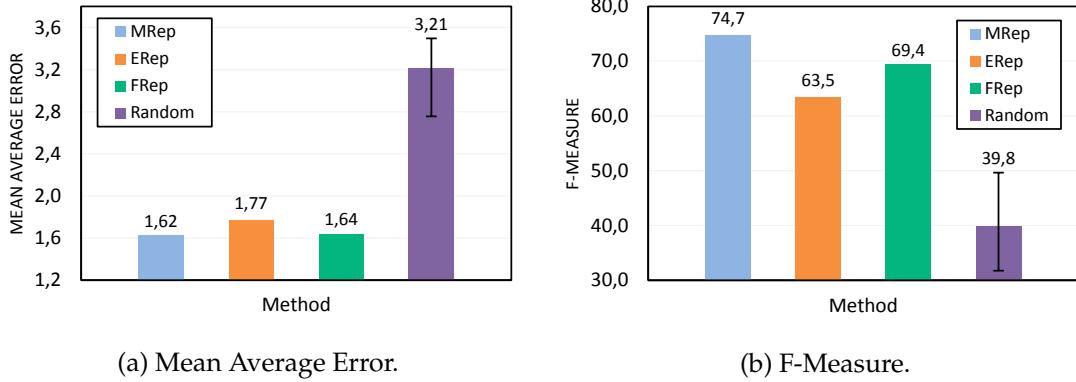


Figure 5.13: Comparative results for user cold-start.

5.4 Summary

This chapter discussed the experimental results and evaluation of the implemented recommendation method, presented in Chapter 4. Publicly available datasets [DDPM13] are not adequate for testing the implemented method. Hence, a dataset was built by crawling IMDb and Twitter: the obtained dataset comprised of 1,064,766 ratings, 52,236 tweets, 124,236 reviews and metadata for 46,903 movies, where 60 were new movies. Rating predictions for the new movies were computed by using the old ratings, tweets, reviews and metadata as the training set, while the real user-movie ratings for the new movies were used as the test set. By comparing various variants of the implemented method to three baseline predictors, the following important conclusions were drawn:

- The best estimated α_t parameter for Equation 4.20 was 0.35, suggesting that the popularity $pop(m_j)$ of a movie m_j is important to predict its user-movie ratings, but should not be considered too much against user preferences;
- The method outperformed the best baseline by an F-Measure of 8.1%, obtaining a total of 87.7%. The method obtained the best results when considering the full spectrum of Social-Media feedback, i.e. both the popularity of the movie and the reputation of its entities;
- The method outperformed random recommendations, a common baseline in literature [LVLD08; PC09], for cases of extreme movie and user *cold-start*, where there are no past ratings. The best approach leveraged only on the popularity of movies and outperformed the best run of random recommendations by 14% of F-Measure, obtaining a total of 74.7%.



Conclusions

This final chapter wraps up the dissertation by presenting its final conclusions. First, a summary of the main contributions and results of the work is presented. The challenges and limitations associated to the implemented framework are then briefly discussed. By the end of the chapter and dissertation, future and complementary work is proposed as a course for maturing the tackled theme and developed work.

6.1 Summary of Contributions

This thesis addressed the problem of recommending new movies, affected by the Cold-Start problem, by monitoring feedback shared on Social-Media platforms. More specifically, it focused on exploring the popularity of new movies, together with the reputation of the corresponding directors and actors, to obtain qualitative measures regarding the quality of candidate new movies. The developed work culminated into a recommendation framework with the following characteristics:

- Starts with a set of candidate new movies and corresponding metadata, where none of the movies has any associated ratings;
- Given a set of tweets, tweets referring the new movies are identified, classified with a k -NN sentiment classifier and indexed by movie title, allowing fast look-ups;
- Given a set of IMDb reviews, the reputations of the directors and actors of the new movies are calculated with an external framework [PSM14a] and indexed according to the corresponding movie titles, allowing fast look-ups;

- Given a set of user-movie ratings, crawls the metadata of the rated movies and correlates it with the respective ratings to discover user preferences and standards;
- Given the name of a user, explores its preferences together with indexed tweets and reputations to recommend new movies, by predicting the corresponding user-movie ratings.

Experiments were performed to assess the various components of the implemented framework, by leveraging on a crawled dataset containing 52,236 tweets, 124,236 IMDb reviews and 1,064,766 IMDb ratings given by 2,909 users to 60 new movies and 46,843 old movies. For the Social-Media Monitoring module it was assessed: (1) the accuracy of the reputation framework [PSM14a] for various popular directors and actors and (2) the accuracy of the tweets classifier in labelling the sentiment on tweets regarding new movies. The reputation framework [PSM14a] presented great results and outperformed the strongest baseline by an average accuracy of 11.8%. The tweets classifier presented good results as well by obtaining an average accuracy of 79.3%.

In turn, for the Cold-Start Recommendation module it was determined: (1) the obtained error when estimating the popularity of new movies, (2) the influence of including user bias for modelling social-media signals to the user standards, (3) the obtained improvement by considering the reputation of directors and actors when modelling user preferences, (4) the obtained improvement when considering the popularity of movies for predicting user-movie ratings and (5) the obtained results when tackling extreme cases of both movie-side and user-side Cold-Start. The method for estimating the popularity of new movies presented very similar results to the best found in literature [OBTR12]. When recommending, the inclusion of user bias for modelling social-media signals improved recommendation F-Measure by an average of 1.6%. Furthermore, modelling the user preferences with the reputation of directors and actors improved over the best baseline by an F-Measure of 6.3%, while modelling the predicted ratings with the popularity of the target movie improved by an F-Measure of 7.6%. By combining both social-media signals, the method outperformed the best baseline by an F-Measure of 8.1%. On extreme cases of both user and movie Cold-Start, the best results outperformed random recommendations by an average F-Measure of 34.9%.

This work also contributed for two scientific papers [PSM14b; PSM14a], published at the 2014 editions of *SIGIR* and *International Conference on Web Intelligence*, which are related to the framework used to compute the reputation of directors and actors. The main contribution comprised the classification of a Ground-Truth dataset, via Crowdsourcing, for evaluating the method and the evaluation of the domain-specific lexicon.

6.2 Challenges and Limitations

Recommendation frameworks are usually oriented into environments that involve user consumption and classification of consumed products. Hence, both the developed work

and resulting framework have potential usefulness for real multimedia mining platforms, such as Amazon and IMDb, where personalized product recommendations are performed. The implemented framework, however, holds several challenges and limitations that would need to be tackled, in order to be truly useful on a realistic scenario. Furthermore, the main challenges and limitations associated with the implemented frameworks are the following:

- The framework starts with a static set of new movies. On a realistic scenario, new movies are released every week and the framework lacks that real-time adaptiveness, as it would ideally update the set of candidate movies periodically;
- A set of static, previously crawled tweets are exploited to infer the reputation of new movies while, ideally, tweets would be crawled and classified in real-time or periodically;
- A set of statics, previously crawled IMDb reviews are used to calculate the reputation of directors and actors while, ideally, both the crawled reviews and the reputations would be updated periodically;
- The time complexity of processing all the various aspects of the framework is not taken in account and therefore optimized for real-life usage, where recommendations need to be computed swiftly. Ideally, the various processing tasks would be scheduled and optimized for fast recommendations.

6.3 Future Work

While the goal of this dissertation is thoroughly fulfilled, this theme and work, as a whole, still has room for major exploration and improvement. Therefore, future work is proposed as the next step not only to tackle the limitations of the current framework, but also to enrich it by exploring other information and techniques:

- **Real-Time, Automatic Candidate New Movies Selection:** The first limitation of the implemented framework is that it considers a static set of manually selected candidate new movies. This limitation can be tackled by updating the list of candidate new movies weekly, as new movies are released every week and those are the most potential victims of the new item *cold-start* problem. A possible approach is to crawl the IMDb section regarding the upcoming movie releases.
- **Real-Time, Automatic Social-Media Mining:** The implemented framework computes the popularity of new movies and the reputation of directors and actors from static tweets and reviews. The logical next step regarding this is to compute this information in real-time, by updating both the set of tweets and the set of reviews periodically. A possible approach is to capture real-time tweets via the Twitter API,

capture reviews on a weekly basis and update the popularities and reputations weekly with the updated sets.

- **Temporal Fluctuations for Preferences and Reputations:** The implemented framework does not consider *time* as a variable. However, in real life, various variables change over time, such as the preferences of a user or the reputation of directors and actors. Hence, the framework can be improved by introducing this concept. For example, user preferences can be estimated by considering newer ratings more relevant when building the user profile, while the reputation of directors and actors can, similarly, consider sentiment expressed on newer reviews to weight more when calculating reputations.
- **Social-Media for Describing New Movies:** Social-Media users do not always refer to new movies in a positive or negative fashion. Sometimes, users share posts where characteristics of new movies are described instead, like the setting of the movie or if it has lots of chase scenes. This information can be useful for better describing new movies. A possible approach to explore this information is to extend the user profile to store characteristics identified on the rated movies and compare these with the characteristics of new movies to better model the similarity rating of users and new movies.
- **Multi-Source Social-Media Trends and Reputations:** There are various social-media platforms besides Twitter and IMDb, and all those collect great amounts of feedback on new movies, director and actors. Exploring various sources not only tackles possible data sparsity but also enables the framework to capture more variety of data and in a much faster manner. Facebook is an example of a possible source of feedback on new movies, as it is a very popular Social Network where users usually share small posts in a similar fashion to Twitter. Youtube¹, in turn, is a different social-media approach, where users post videos and other users can comment or rate with a *like/dislike*. A possible approach is, for example, to capture feedback targeting directors and actors from comments on videos regarding those entities or regarding movies where they have participated.

¹www.youtube.com

Bibliography

- [AZSD07] S. Aciar, D. Zhang, S. Simoff, and J. Debenham. “Informed recommender: Basing recommendations on consumer product reviews”. In: *Intelligent Systems, IEEE* 22.3 (2007), pp. 39–47.
- [AT05] G. Adomavicius and A. Tuzhilin. “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. In: *Knowledge and Data Engineering, IEEE Transactions on* 17.6 (2005), pp. 734–749.
- [ASS13] G. Alexandridis, G. Siolas, and A. Stafylopatis. “Improving Social Recommendations by applying a Personalized Item Clustering Policy.” In: *RSWeb@ RecSys*. Citeseer. 2013.
- [ALPKO09] X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak, and N. Oliver. “The wisdom of the few: a collaborative filtering approach based on expert opinions from the web”. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2009, pp. 532–539.
- [BS97] M. Balabanović and Y. Shoham. “Fab: content-based, collaborative recommendation”. In: *Communications of the ACM* 40.3 (1997), pp. 66–72.
- [BHC+98] C. Basu, H. Hirsh, W. Cohen, et al. “Recommendation as classification: Using social and content-based information in recommendation”. In: *AAAI/IAAI*. 1998, pp. 714–720.
- [BC92] N. J. Belkin and W. B. Croft. “Information filtering and information retrieval: two sides of the same coin?” In: *Communications of the ACM* 35.12 (1992), pp. 29–38.
- [BCHC09] J. Benesty, J. Chen, Y. Huang, and I. Cohen. “Pearson correlation coefficient”. In: *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.
- [BP00] D. Billsus and M. J. Pazzani. “User modeling for adaptive news access”. In: *User modeling and user-adapted interaction* 10.2-3 (2000), pp. 147–180.

- [BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent dirichlet allocation". In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.
- [BHK98] J. S. Breese, D. Heckerman, and C. Kadie. "Empirical analysis of predictive algorithms for collaborative filtering". In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1998, pp. 43–52.
- [BGC10] A. Brew, D. Greene, and P. Cunningham. "Using Crowdsourcing and Active Learning to Track Sentiment in Online Media." In: *ECAI*. 2010, pp. 145–150.
- [Bur02] R. Burke. "Hybrid recommender systems: Survey and experiments". In: *User modeling and user-adapted interaction* 12.4 (2002), pp. 331–370.
- [BV13] R. Burke and F. Vahedian. "Social Web Recommendation using Metaphaths." In: *RSWeb@ RecSys*. Citeseer. 2013.
- [CWNWS12] L. Chen, W. Wang, M. Nagarajan, S. Wang, and A. P. Sheth. "Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter." In: *ICWSM*. 2012.
- [CGMMNS99] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. "Combining content-based and collaborative filters in an online newspaper". In: *Proceedings of ACM SIGIR workshop on recommender systems*. Vol. 60. Citeseer. 1999.
- [CS00] P. Cotter and B. Smyth. "Ptv: Intelligent personalised tv guides". In: *AAAI/IAAI*. 2000, pp. 957–964.
- [DC01] S. Das and M. Chen. "Yahoo! for Amazon: Sentiment parsing from small talk on the Web". In: *EFA 2001 Barcelona Meetings*. 2001.
- [DS10] N. A. Diakopoulos and D. A. Shamma. "Characterizing debate performance via aggregated twitter sentiment". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2010, pp. 1195–1198.
- [DDPM13] S. Dooms, T. De Pessemier, and L. Martens. "Movietweetings: a movie rating dataset collected from twitter". In: *Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys*. Vol. 2013. 2013.
- [ESK03] L. Ertoz, M. Steinbach, and V. Kumar. "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data". In: *Proc. Third SIAM Intl Conf. Data Min.* 2003, p. 47.
- [ES06] A. Esuli and F. Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining". In: *Proceedings of LREC*. Vol. 6. 2006, pp. 417–422.

- [FS07] R. Feldman and J. Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007.
- [FMKKMD10] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. "Annotating named entities in Twitter data with crowdsourcing". In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics. 2010, pp. 80–88.
- [FA99] H. H. Friedman and T. Amoo. "Rating the rating scales". In: *Journal of Marketing Management* 9.3 (1999), pp. 114–123.
- [GM05] E. Gabrilovich and S. Markovitch. "Feature generation for text categorization using world knowledge". In: *IJCAI*. Vol. 5. 2005, pp. 1048–1053.
- [GNOT92] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. "Using collaborative filtering to weave an information tapestry". In: *Communications of the ACM* 35.12 (1992), pp. 61–70.
- [GGMS13] D. F. Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti. "A Sentiment-Based Approach to Twitter User Recommendation." In: *RSWeb@ RecSys*. 2013.
- [HMM13] B. Haslhofer, F. Martins, and J. Magalhães. "Using SKOS vocabularies for improving web search". In: *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee. 2013, pp. 1253–1258.
- [HSRF95] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. "Recommending and evaluating choices in a virtual community of use". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, Addison-Wesley Publishing Co. 1995, pp. 194–201.
- [Hof99] T. Hofmann. "Probabilistic latent semantic indexing". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 1999, pp. 50–57.
- [Hof04] T. Hofmann. "Latent semantic models for collaborative filtering". In: *ACM Transactions on Information Systems (TOIS)* 22.1 (2004), pp. 89–115.
- [HMS09] P.-Y. Hsueh, P. Melville, and V. Sindhvani. "Data quality from crowdsourcing: a study of annotation selection criteria". In: *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*. Association for Computational Linguistics. 2009, pp. 27–35.
- [HL04] M. Hu and B. Liu. "Mining and summarizing customer reviews". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2004, pp. 168–177.

- [JWMG09] N. Jakob, S. H. Weber, M. C. Müller, and I. Gurevych. "Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations". In: *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. ACM. 2009, pp. 57–64.
- [Jen96] F. V. Jensen. *An introduction to Bayesian networks*. Vol. 210. UCL press London, 1996.
- [JO11] Y. Jo and A. H. Oh. "Aspect and sentiment unification model for online review analysis". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM. 2011, pp. 815–824.
- [KMMHGR97] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. "GroupLens: applying collaborative filtering to Usenet news". In: *Communications of the ACM* 40.3 (1997), pp. 77–87.
- [KNSFG08] J. Krauss, S. Nann, D. Simon, K. Fischbach, and P. Gloor. "Predicting movie success and academy awards through sentiment and social network analysis". In: (2008).
- [LVLD08] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. "Addressing cold-start problem in recommendation systems". In: *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. ACM. 2008, pp. 208–211.
- [LCC06] C. W. Leung, S. C. Chan, and F.-I. Chung. "Integrating collaborative filtering and sentiment analysis: A rating inference approach". In: *Proceedings of The ECAI 2006 Workshop on Recommender Systems*. Citeseer. 2006, pp. 62–66.
- [LSD12] C. Li, A. Sun, and A. Datta. "Twevent: Segment-based event detection from tweets". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM. 2012, pp. 155–164.
- [LSY03] G. Linden, B. Smith, and J. York. "Amazon. com recommendations: Item-to-item collaborative filtering". In: *Internet Computing, IEEE* 7.1 (2003), pp. 76–80.
- [Lit88] N. Littlestone. "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm". In: *Machine learning* 2.4 (1988), pp. 285–318.
- [LGS11] P. Lops, M. de Gemmis, and G. Semeraro. "Content-based recommender systems: State of the art and trends". In: *Recommender Systems Handbook*. Springer, 2011, pp. 73–105.

- [MWGA13] T. Martín-Wanton, J. Gonzalo, and E. Amigó. "An unsupervised transfer learning approach to discover topics for online reputation management". In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM. 2013, pp. 1565–1568.
- [MPM12] F. Martins, F. Peleja, and J. Magalhães. "SentiTVchat: sensing the mood of social-TV viewers". In: *Proceedings of the 10th European conference on Interactive tv and video*. ACM. 2012, pp. 161–164.
- [MK10] M. Mathioudakis and N. Koudas. "Twittermonitor: trend detection over the twitter stream". In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM. 2010, pp. 1155–1158.
- [MMN02] P. Melville, R. J. Mooney, and R. Nagarajan. "Content-boosted collaborative filtering for improved recommendations". In: *AAAI/IAAI*. 2002, pp. 187–192.
- [MBW07] R. Mihalcea, C. Banea, and J. Wiebe. "Learning multilingual subjective language via cross-lingual projections". In: *Annual Meeting, Association for Computational Linguistics*. Vol. 45. 1. 2007, p. 976.
- [MCHLM08] S. Milstein, A. Chowdhury, G. Hochmuth, B. Lorica, and R. Magoulas. *Twitter and the Micro-blogging Revolution*. 2008.
- [MT12] S. M. Mohammad and P. D. Turney. "Crowdsourcing a word–emotion association lexicon". In: *Computational Intelligence* (2012).
- [MPJ11] Y. Moshfeghi, B. Piwowarski, and J. M. Jose. "Handling data sparsity in collaborative filtering using emotion and semantic based features". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM. 2011, pp. 625–634.
- [NR10] S. Nowak and S. Rüger. "How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation". In: *Proceedings of the international conference on Multimedia information retrieval*. ACM. 2010, pp. 557–566.
- [OK+98] D. W. Oard, J. Kim, et al. "Implicit feedback for recommender systems". In: *Proceedings of the AAAI workshop on recommender systems*. Wollongong. 1998, pp. 81–83.
- [OBTR12] A. Oghina, M. Breuss, M. Tsagkias, and M. de Rijke. "Predicting imdb movie ratings using social media". In: *Advances in information retrieval*. Springer, 2012, pp. 503–507.
- [OSO12] O. Ozdakis, P. Senkul, and H. Oguztuzun. "Semantic expansion of hashtags for enhanced event detection in twitter". In: *Proceedings of the 1st International Workshop on Online Social Systems*. 2012.

- [PL04] B. Pang and L. Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts". In: *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2004, p. 271.
- [PL05] B. Pang and L. Lee. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2005, pp. 115–124.
- [PL08] B. Pang and L. Lee. "Opinion mining and sentiment analysis". In: *Foundations and trends in information retrieval* 2.1-2 (2008), pp. 1–135.
- [PLV02] B. Pang, L. Lee, and S. Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics. 2002, pp. 79–86.
- [PC09] S.-T. Park and W. Chu. "Pairwise preference regression for cold-start recommendation". In: *Proceedings of the third ACM conference on Recommender systems*. ACM. 2009, pp. 21–28.
- [Paz99] M. J. Pazzani. "A framework for collaborative, content-based and demographic filtering". In: *Artificial Intelligence Review* 13.5-6 (1999), pp. 393–408.
- [PDM12] F. Peleja, P. Dias, and J. Magalhães. "A Regularized Recommendation Algorithm with Probabilistic Sentiment-Ratings". In: *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 701–708.
- [PSM14a] F. Peleja, J. Santos, and J. Magalhães. "Ranking Linked-Entities in a Sentiment Graph". In: *International Conference on Web Intelligence*. WIC. 2014.
- [PSM14b] F. Peleja, J. Santos, and J. Magalhães. "Reputation analysis with a ranked sentiment-lexicon". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM. 2014, pp. 1207–1210.
- [PPL01] A. Popescul, D. M. Pennock, and S. Lawrence. "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments". In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 2001, pp. 437–444.
- [RDL10] D. Ramage, S. T. Dumais, and D. J. Liebling. "Characterizing Microblogs with Topic Models." In: *ICWSM*. 2010.

- [Sal89] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*. Addison-Wesley, 1989.
- [SKKR00] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. *Application of dimensionality reduction in recommender system-a case study*. Tech. rep. DTIC Document, 2000.
- [SM95] U. Shardanand and P. Maes. “Social information filtering: algorithms for automating “word of mouth””. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co. 1995, pp. 210–217.
- [SPI08] V. S. Sheng, F. Provost, and P. G. Ipeirotis. “Get another label? improving data quality and data mining using multiple, noisy labelers”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2008, pp. 614–622.
- [SM93] B. Sheth and P. Maes. “Evolving agents for personalized information filtering”. In: *Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on*. IEEE. 1993, pp. 345–352.
- [SOJN08] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng. “Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks”. In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2008, pp. 254–263.
- [SN99] I. Soboroff and C. Nicholas. “Combining content and collaboration in text filtering”. In: *Proceedings of the IJCAI*. Vol. 99. 1999, pp. 86–91.
- [SAMAGG13] D. Spina, J. Carrillo-de Albornoz, T. Martín, E. Amigó, J. Gonzalo, and F. Giner. “UNED Online Reputation Monitoring Team at RepLab 2013”. In: CLEF. 2013.
- [SM08] C. Strapparava and R. Mihalcea. “Learning to identify emotions in text”. In: *Proceedings of the 2008 ACM symposium on Applied computing*. ACM. 2008, pp. 1556–1560.
- [TMM13] G. Tavares, A. Mourão, and J. Magalhaes. “Crowdsourcing for affective-interaction in computer games”. In: *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*. ACM. 2013, pp. 7–12.
- [TC00] T. Tran and R. Cohen. “Hybrid recommender systems for electronic commerce”. In: *Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04*, AAAI Press. 2000.

- [Tur02] P. D. Turney. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 417–424.
- [UF98] L. H. Ungar and D. P. Foster. "Clustering methods for collaborative filtering". In: *AAAI Workshop on Recommendation Systems*. 1. 1998.
- [WM12] S. Wang and C. D. Manning. "Baselines and bigrams: Simple, good sentiment and topic classification". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics. 2012, pp. 90–94.
- [WWH05] T. Wilson, J. Wiebe, and P. Hoffmann. "Recognizing contextual polarity in phrase-level sentiment analysis". In: *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics. 2005, pp. 347–354.
- [YCK09] M.-C. Yuen, L.-J. Chen, and I. King. "A survey of human computation systems". In: *Computational Science and Engineering, 2009. CSE'09. International Conference on*. Vol. 4. IEEE. 2009, pp. 723–728.
- [YKL11a] M.-C. Yuen, I. King, and K.-S. Leung. "A survey of crowdsourcing systems". In: *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*. IEEE. 2011, pp. 766–773.
- [YKL11b] M.-C. Yuen, I. King, and K.-S. Leung. "Task matching in crowdsourcing". In: *Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*. IEEE. 2011, pp. 409–412.
- [ZDCL10] W. Zhang, G. Ding, L. Chen, and C. Li. "Augmenting chinese online video recommendations by using virtual ratings predicted by review sentiment classification". In: *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1143–1150.